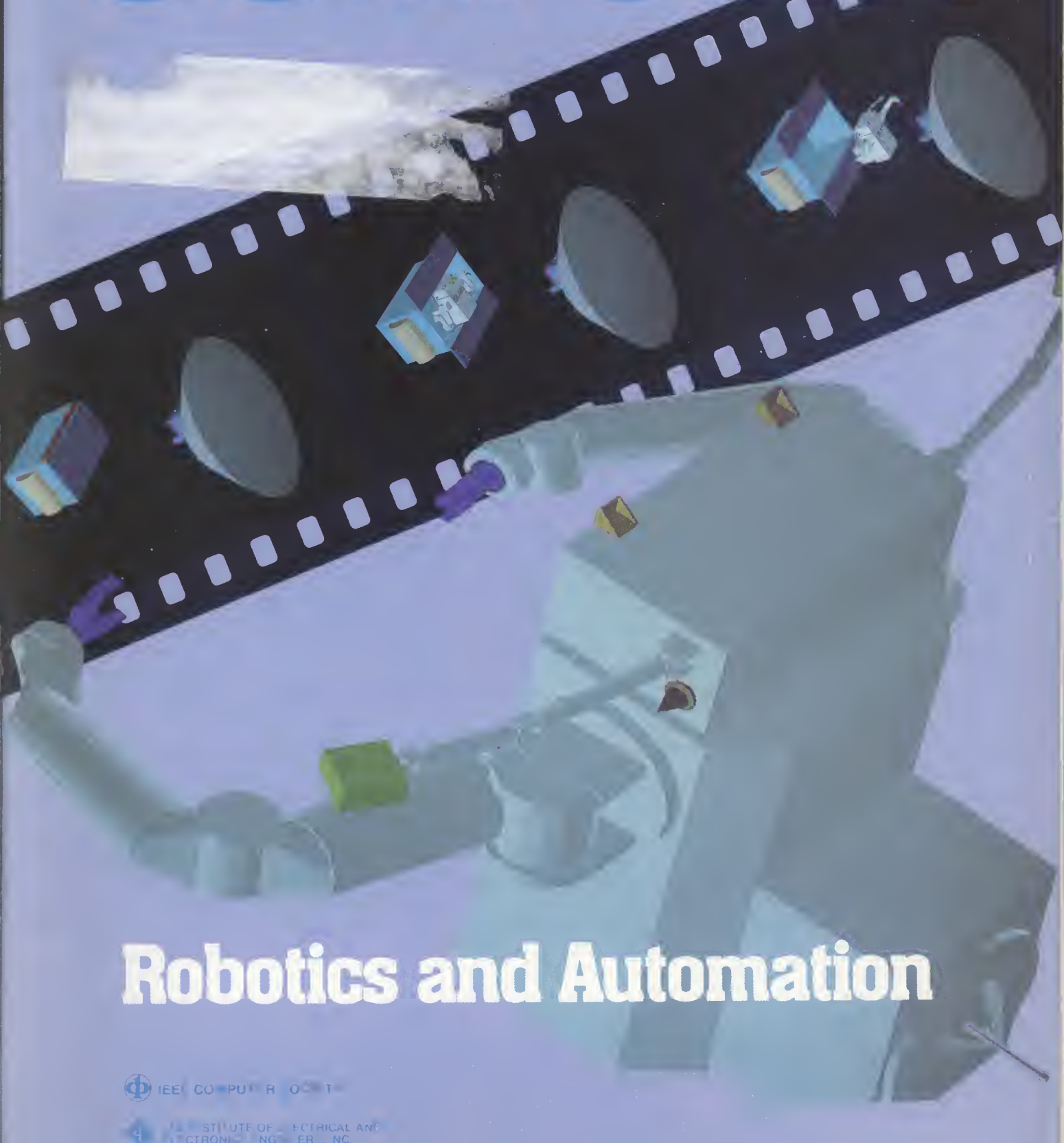


MARCH 1989

COMPUTER



Robotics and Automation

IEEE COMPUTER SOCIETY

4 INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERING, INC.



IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN

ICCAD-89

A CONFERENCE FOR THE
EE CAD PROFESSIONAL

NOVEMBER 6 - 9, 1989
SANTA CLARA, CA

sponsored by
IEEE CIRCUITS AND SYSTEMS SOCIETY
THE IEEE COMPUTER SOCIETY

CALL FOR PAPERS

The 1989 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN will be held November 6 - 9, 1989. The Conference oriented towards Electrical Engineering CAD professionals, concentrating on CAD for Electronic Circuit Design.

AREAS OF INTEREST

Original technical papers on (but not limited to) the following topics are invited:

- 1) **HIGH LEVEL SIMULATION:** Functional, Behavioral, Architectural, Mixed-mode, Fault Simulation, Hardware Acceleration and Parallel Techniques for High Level Simulation
- 2) **SIMULATION:** Timing, Circuit, Device and Process Simulation, Modeling, Hardware Acceleration and Parallel Techniques for Simulation
- 3) **PLACEMENT AND FLOORPLANNING:** Placement, Floorplanning, Area Estimation, Layout Systems, Hardware Acceleration and Parallel Techniques
- 4) **ROUTING:** Routing algorithms for IC and System Design
- 5) **LAYOUT VERIFICATION/EXTRACTION:** Circuit Extraction/Verification, Design Rule Checking and Electrical Rule Checking, Symbolic Design and Compaction
- 6) **TESTING:** Design for Testability, Automatic Test Pattern Generation, Built-in Test
- 7) **LOGIC AND ANALOG SYNTHESIS:** Logic Synthesis and Formal Verification, Technology Mapping, Finite-State Machine Synthesis, Analog Circuit synthesis
- 8) **HIGH LEVEL SYNTHESIS:** Functional, Behavioral, Architectural Synthesis, Synthesis Systems, Silicon Compilation
- 9) **CAD FRAMEWORKS:** Tool Integration, User-interfaces, Data-bases for IC and system design.

AUTHOR INFORMATION

Authors should submit 12 COPIES of:

- 1) a one-paragraph abstract.
- 2) a more detailed description not to exceed 18 double-spaced pages, figures and tables included. Excessively long submissions and previously published papers will be returned to the authors.

FORMAT:

The ONE-PARAGRAPH ABSTRACT, typed on one separate page, should clearly and precisely state what is new and point out the significant results. Succinctness is required since this paragraph may be included in the Advance Program.

In the detailed description, the author must objectively address why the proposed contribution is superior to prior work or what the significance of the contribution is, if breaking new ground. Demonstration of superiority in algorithms and strategies with heuristics is required through a description of the programming implementation and application to "real" problems. Additional mathematical proofs are welcome. The contribution should address an area of current technical interest to the CAD professional. A clear description of the new contribution, status of the work and significant examples and results should be given.

COVER PAGE REQUIREMENTS:

Submissions should include, on the cover page: the title of the paper; the category which most clearly matches the paper's content (see Areas of Interest); the full name, complete return address, telephone number and affiliation of each author; and clear identification of the individual to whom all communication should be addressed. In giving your return address, please consider that the communications for paper acceptance and mailing of the author's kit occur in the month of July.

AUTHOR'S SCHEDULE

Deadline for submissions:	April 28, 1989
Notification of acceptance:	June 30, 1989
Deadline for final version:	August 11, 1989

SEND TO:

ICCAD-89 Secretary
IEEE Computer Society
1730 Massachusetts Ave. N.W.
Washington, D.C. 20036-1903
Telephone: (202) 371-1013

GENERAL CHAIRPERSON

Andrzej J. Strojwas
ECE Department
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 268-3530

PROGRAM CHAIRPERSON

Alberto Sangiovanni-Vincentelli
Department of EE and CS
University of California, Berkeley
Berkeley, CA 94720
(415) 642-4882

PUBLICATIONS CHAIRPERSON

Satoshi Goto
C and C Information Technology Research Labs
NEC Corporation
1-1 Miyazaki 4-Chome
Miyamae-ku, Kawasaki
Kanagawa-213, Japan
011-81-44-856-2170

NEW FOR NETWORK DESIGNERS

BEFORE

```

/* PLACE ENTITY IN WAITING LINE */
IF NQUE=0 THEN FIRST=P1
ELSE LAST=FOUR+P1
CUMD=CUMD+NQUE*(CLOCK-FLAST);
FLAST=CLOCK;
NQUE=NQUE+1;
NMAX=NMAX+NQUE;
LAST=P1
TNAT_ARY=CLOCK*ENDIRATE_EACH_RATE_TOT;
RETURN;
END;

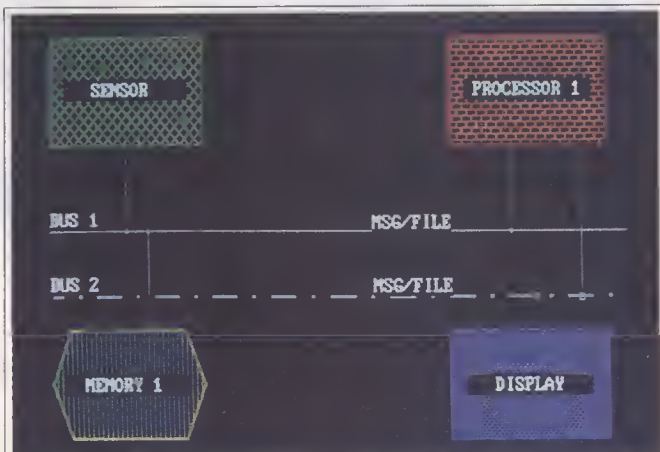
/* DEPARTURE ROUTINE */
PROCEDURE;
/* FREE BOOTH */
CTEL=CTEL+TEL->USE_TIME;
STATE=01;
FREE TEL->USER;
TE NQUE=0 THEN DO;
/* SYSTEM IS EMPTY, FORCE NEXT */
/* EVENT TO BE ARRIVAL */
TNAT_OPRT=150;
RETURN;
END;
CUMD=CUMD+NQUE*(CLOCK-FLAST);
FLAST=CLOCK;
CUMT=CUMT+ (CLOCK-ETEST->ARRV_TIME);
/* PLACE FIRST WAITING CALLER IN BOOTH */
TEL=FIRST;
ETEST=TEL->EOLL;
NQUE=NQUE-1;
NMAX=NMAX-1;
STATE=1;
TNAT_OPRT=CLOCK+TEL->USE_TIME;
RETURN;
END;

/* RANDOM NUMBER GENERATOR */
PROCEDURE (RNDM,TOT);
DECL FPRD=0; FTEST=0;
DECL (L1=1); STATE=0; BTH=0; THIT=1;
CALL RANDUM(TY,VPL);
TE=TY;
NFRAGMVEL=ITC+T-ERDM;
RETURN (M);
END;

```

Costs and delays of programming

AFTER



Computer-communications network simulation
--quick results, no programming

See your proposed network perform under various workloads

NETWORK II.5 now gives you easy-to-understand simulation results quickly -- no programming

With NETWORK II.5 you enter your computer-communications network description.

Simulation follows immediately --no programming delays.

Easy-to-understand results

Your reports show utilization, queues, execution times, response times, and conflicts.

Graphical reports show hardware layout, software data flow, and device utilization.

Seeing graphical results increases everyone's understanding of your proposed network and builds confidence in the analysis.

Your system simulated

You can analyze any computer-communication system including local area networks.

You can simulate some portions of the system at a detailed level and others at a coarser level.

Widely used protocols are built-in --you just make a choice.

Free trial and training offer

The free trial contains everything you need to try NETWORK II.5® on your computer.

You may develop your own network or modify one of ours.

Try the NETWORK II.5 approach, the quality and timeliness of our support, the accuracy of our documentation, and the facilities for error checking -- **everything you need for a successful project.**

For 26 years CACI has offered trial use of its simulation software -- **no cost, no obligation.**

Act now -- free training offer

For a limited time we also include **free training.** Typical applications are demonstrated there.

Call today to avoid disappointment -- class size is limited.

For immediate information

Call Paul Gorman at (619) 457-9681, FAX (619) 457-1184. In the UK, call Richard Eve on (01) 528-7980.

The quickest, easiest way for you to evaluate network alternatives is with NETWORK II.5.

Rush information on the NETWORK II.5 free trial and training offer

Free trial -- no cost, no obligation.

Limited offer -- Act now for free training.

Also send information on:

☐ COMNET II.5-Telecommunication network analysis without programming.

☐ Special University Offer.

Name _____

Organization _____

Address _____

City _____

State _____

Zip _____

Telephone _____

Computer _____

IEEE COMP

Return to:

CACI Products Company
3344 North Torrey Pines Court
La Jolla, California 92037
Call Paul Gorman at (619) 457-9681,
or FAX (619) 457-1184

In the UK

CACI Products Division
Regent House, 89 Kingsway
London WC2B 6RH, United Kingdom
Call Richard Eve on (01) 528-7980

COMPUTER

March 1989

Published by the IEEE Computer Society

Vol. 22 No. 3

ARTICLES

6 Guest Editor's Introduction: Robotics and Automation

Gordon T. Wilfong

9 Algorithmic Motion Planning in Robotics

Micha Sharir

An intelligent tool, or a robot that thrashes about uselessly in an environment it cannot comprehend? The difference may lie in the painstaking development of geometric algorithms.

21 Task-Level Planning of Pick-and-Place Robot Motions

Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell

The Handey system breaks the pick-and-place problem into nearly independent, computationally feasible subproblems as a step toward a comprehensive task-level system.

31 The Problems of Accuracy and Robustness in Geometric Computation

Christoph M. Hoffmann

Despite the pressing need, devising accurate and robust geometric algorithms has proved elusive for many important science and engineering applications. Why is that? And, how can we change it?

43 Exception Handling in Robotics

Ingemar J. Cox and Narain H. Gehani

Errors caused by abnormal external events can cause logically correct robot programs to fail. Exception handling facilities provide a mechanism for writing reliable, robust, and maintainable robot programs.

50 Current Research in Robotics and Automation

Four overviews provide a sampling of current research in robotics. Included are reports from Columbia University, Sandia National Laboratories, Cornell University, and the University of Southern California.

SPECIAL FEATURE

63 Secure Query-Processing Strategies

T.F. Keefe, M.B. Thuraisingham, and W.T. Tsai

These techniques allow users with different clearance levels to access data at various sensitivity levels without violating the security of a database management system.

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, phone (714) 821-8380. IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; IEEE Headquarters, 345 East 47th St., New York, NY 10017. Annual subscription included in society member dues. Nonmember subscription rates: available upon request. Single copy prices: members \$10.00; nonmembers \$20.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Second class postage is paid at New York, New York, and at additional mailing offices.

Copyright and reprint permission: Copyright © 1989 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons: (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970; (2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to Permissions Editor, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space considerations. *Computer* subscribes to the Computer Press Association's code of professional ethics.



DEPARTMENTS



The three frames on the cover are from a longer sequence, provided by Silma, Inc., Cupertino, California. The complete sequence shows a robot opening an enclosure, removing an object from within, and closing the enclosure again. The entire action is featured as a flip sequence beginning on p. 7. See "About the cover," p. 7.

Cover design: Jay Simpson,
Design & Direction

In the next issue

Software Tools for Hardware Test

72 Standards

EISA 32-bit connector finalized

74 Computer Society News

IEEE fellows; TAB report; Gordon Bell Award

77 Update

NSF seeks 14 percent budget boost; Bitnet OKs CSnet merger

81 Product Reviews

Data control, acquisition, analysis, and display

90 New Products

95 IC Announcements

96 Microsystem Announcements

98 Conferences

Supercomputers; robotics and automation; communications; simulation; languages

101 Calendar

105 Call for Papers

116 Book Reviews

119 New Literature

Computer Society Information Page, inside back cover;
Membership Application, 115; Career Opportunities, 108;
Advertiser/Product Index, 120; Reader Service Card, 120A

COMPUTER

10662 Los Vaqueros Circle, Los Alamitos, CA 90720

Editor-in-Chief

Bruce D. Shriver, University of Hawaii

Editorial Board

Dharma P. Agrawal, North Carolina State University

Dennis R. Allison, Stanford University

Scott Davidson, AT&T

Richard Eckhouse, MOCO

Scott E. Fahlman, Carnegie Mellon University

Edmund Gallizzi, Eckerd College

John Gurd, University of Manchester

James H. Haynes, University of California, Santa Cruz

Wiley McKinzie, Rochester Institute of Technology

K.M. Mohiuddin, IBM Almaden Research Laboratory

Yale Patt, University of Michigan

William Riddle, Software Productivity Consortium

Sallie Sheppard, Texas A&M University

Helen M. Wood, National Oceanic and Atmospheric Administration

Akinori Yonezawa, Tokyo Institute of Technology

Submissions

Send six double-spaced copies of articles and special-issue proposals to
Bruce D. Shriver, Dept. of Decision Sciences, University of Hawaii, 2404
Maile Way, Honolulu, HI 96822, phone (808) 948-7231. For electronic sub-
mission, Shriver's addresses are b.shriver on Compmail +, shriver@uhccux
on Bitnet, and shriver@uhccux.uhcc.hawaii.edu.

Staff

Editor and Publisher: True Seaborn

Assistant Publisher: Douglas L. Combs

Assistant to the Publisher: Pat Paulsen

Managing Editor: Marilyn Potes

Assistant Editors: Nancy Hays, Chuck Governale,
Steve H. Wilcox

Contributing Editor: Ware Myers

Art Director: Jay Simpson

Design/Production: Miriam Wiegel

Membership/Circulation: Christina Champion

Advertising Director: Dawn Peck

Advertising Coordinators: Heidi Rex, Marian Tibayan

Magazine Advisory Committee

Sallie Sheppard (acting chair), Sumit DasGupta,

Michael Evangelist, Joe Hootman, Ted Lewis, David Pessel,

True Seaborn, Bruce D. Shriver, John Staudhammer

Publications Board

Sallie Sheppard (chair), James H. Aylor, Victor Basili,

J. Richard Burke, Jon T. Butler, J.T. Cain, Sumit DasGupta,

Michael Evangelist, Joe Hootman, Glen G. Langdon, Ted Lewis,

Ming T. Liu, Theo Pavlidis, David Pessel, Bruce D. Shriver,

Charles B. Silio, John Staudhammer, Harold Stone, Steven L. Tanimoto

ADVANCE ANNOUNCEMENT

1989 IEEE Computer Society Symposium on Security and Privacy

May 1-3, 1989 • Claremont Resort • Oakland, California

MONDAY, MAY 1, 1989

- 8:45-9:00 a.m. **Welcome, Introduction**
Terry Benzel, General Chair
Tom Berson, Program Co-Chair
- 9:00-10:30 a.m. **Fundamentals Revisited**
Tom Berson, Chair
Network Security: The Parts Of The Sum,
Stephen T. Walker
Statistical Models of Trust: TCB's vs. People,
Theodore M. P. Lee
Symbol Security Condition Considered
Harmful, *Marvin Schaefer*
- 10:30-11:00 a.m. Break
- 11:00 a.m.-12 noon **Operating Systems 1**
Steve Lipner, Chair
New Methods for Immediate Revocation, *Paul A. Karger*
An Identity-Based Capability System That Can
Enforce Security Policies, *Li Gong*
- 12 noon-1:30 p.m. Lunch
- 1:30-3:00 p.m. **Operating Systems 2**
Virgil Gligor, Chair
Access Mediation in a Message Passing
Kernel, *Martha Branstad, Homayoon Tajalli*
Frank Mayer, David Dalva
A Formal Model For Unix Setuid, *Tim Levin,*
Steven J. Padilla, Cynthia E. Irvine
Policy vs. Mechanism in the Secure TUNIS
Operating System, *Guy-L. Grenier, Richard*
C. Holt
The Incorporation of Multi-Level IPC Into
Unix, *Thomas J. Parenty*
- 3:00-3:30 p.m. Break
- 3:30-5:00 p.m. **Database Management Systems**
Catherine Meadows, Chair
Aggregation and Inference: Facts and Fallacies,
Teresa F. Lunt
A Security Model for Object-Oriented Data-
bases, *Eduardo B. Fernandez, Ehud Gudes,*
Haiyan Song
A Security Policy for a Multilevel Secure
Relational Database Management System
with Rows, *Jackson Wilson*
On the Cell Suppression by Merging
Technique in the Lattice Model of Summary
Tables, *Tzong-An Su, JiYong Chung,*
Gultekin Ozsoyoglu
- 5:30-7:30 p.m. **Reception**
- 8:00 p.m.-12 midnight **Poster Sessions, Conversation Room**

TUESDAY, MAY 2, 1989

- 9:00-10:30 a.m. **Praxis**
Earl Boebert, Chair
Using Narrowing in the Analysis of Key
Management Protocols, *Catherine Meadows*
A Software Engineering Approach to Designing
Trustworthy Software, *Mark R. Cornwell*
Formal Model of a Trusted File Server, *John*
C. Williams, George W. Dinolt
LOCK Trek: Navigating Uncharted Space, *O.*
Sami Saydjari, Joseph M. Beckman, Jeffrey
R. Leaman
- 10:30-11:00 a.m. Break

- 11:00 a.m.-12 noon **Distributed Systems**
Deborah Cooper, Chair
Special Mystery Presentation—Don't Miss It!
Security Issues in Policy Routing, *Deborah*
Estrin, Gene Tsudik
The Hierarchical Model of Distributed System
Security, *Glenn Benson, Bill Appelbe, Ian*
Akyildiz
- 12 noon-1:30 p.m. Lunch
- 1:30-3:00 p.m. **Formal Models 1**
John McHugh, Chair
The Chinese Wall Security Policy, *David F. C.*
Brewer, Michael Nash
A 'New' Security Policy Model, *Phil Terry,*
Simon Wiseman
A Framework for Expressing Models of
Security Policy, *John E. Dobson, John A.*
McDermid
- 3:00-3:30 p.m. Break
- 3:30-5:00 p.m. **Formal Models 2**
John McLean, Chair
On The Derivation of Secure Components,
Jeremy Jacob
A Model for Secure Information Flow, *Simon*
N. Foley
Transformation of Access Rights, *Ravi Sandhu*
A Model for Specifying Multi-Granularity
Integrity Policies, *Lee Badger*
- 5:00-6:00 p.m. **Technical Committee Business Meeting**
- 8:00 p.m.-12 midnight **Poster Sessions, Conversation Room**

WEDNESDAY, MAY 3, 1989

- 9:20-10:20 a.m. **Authentication**
Teresa Lunt, Chair
Detection of Anomalous Computer Session
Activity, *H. S. Vaccaro*
Layering Central Authentication on Existing
Distributed System Terminal Services,
Dan Nessett
Authenticated Group Key Distribution Scheme
for a Large Distribution Network, *Lein Ham,*
Thomas Keisler
- 10:20-10:30 a.m. **Presentation of Awards**
- 10:30-11:00 a.m. Break
- 11:00 a.m.-12 noon **Computer Viruses**
Jim Anderson, Chair
Defending Systems Against Viruses through
Cryptographic Authentication, *George I.*
David, Yvo G. Desmedt, Brian J. Matt
A Proposal for a Verification-Based Virus
Filter, *Steve Crocker, Maria Pozzo*
- 12 noon-1:30 p.m. Lunch
- 1:30-3:00 p.m. **Computer Virus Panel**
Tom Berson, Chair
Panel Paper: An Analysis of the Internet
Virus of November 1988, *Mark W. Eichin,*
Jon A. Rochlis
Panel Members: *Jim Anderson, Steve Crocker,*
George David, Mark Eichin
- 3:00-3:15 p.m. **Closing Remarks**
- 3:15 p.m. **Symposium Adjourns**

ADVANCE ANNOUNCEMENT

Advance Registration • Register Now!

Complete this form and return to:

Deborah Downs
The Aerospace Corporation
2350 E. El Segundo Blvd.
El Segundo, CA 90245-4691
(213) 336-3052

Symposium Registration

Advance (2 24 89 to 3 17 89)
Late (3 18 89 to 4 1 89)

☐ Member \$220
☐ Member \$260

☐ Non-Member \$280
☐ Non-Member \$340

☐ Student \$50
☐ Student \$50

Registration includes one copy of the proceedings. Additional copies of the proceedings will be available for \$15 each during the symposium. **Advance Registration is mandatory. There will be no on-site registration. Attendance is limited to 350 persons. No refunds.** There will be no per-organization limits imposed.

Please answer the following questions:

Are you a full-time registered student? ☐ YES ☐ NO

Have you participated in any research, development, or evaluation project in computer security: ☐ YES ☐ NO

If yes, please name the project and the area of computer security:

Can you discuss the results of the project at the symposium (subject to constraints of classification and proprietary information)? ☐ YES ☐ NO

Name _____ Company _____ Mail Stop _____

Street Address _____ City State Zip Country _____

Phone Number (_____) _____ IEEE or IEEE Computer Society Member Number _____

Make check (US Dollars) payable to "IEEE Symposium on Security and Privacy."

Claremont Resort Registration • Register Now!

1989 IEEE Symposium on Security and Privacy • April 30-May 3, 1989

Cut-off date: March 29, 1989

Note: Reservation requests received after the cut-off date will be accepted on a space available basis and are subject to prevailing hotel rates. Please be advised that our check-in time is after 3:00 p.m. and check-out time is 12:00 noon.

Type of accommodations being requested:

☐ Single ☐ Double (2 persons 1 bed) ☐ Double Double (2 persons 2 beds)

All reservations must be accompanied by an advance deposit or credit card guarantee. You may cancel your individual reservations up to 72 hours prior to arrival after which your deposit becomes non-refundable.

To reserve complimentary Oakland Airport Transportation please call 24 hours in advance (415) 843-3000 Ext. 133.

Group Rates: ☐ \$87-Single ☐ \$99-Double*

Signature _____ *Sharing with: (Name) _____

NAME _____

COMPANY/GROUP _____

STREET ADDRESS _____ PHONE (_____) _____

CITY STATE/ZIP _____

I will arrive on (Day) _____ (Date) _____ I will depart on (Day) _____ (Date) _____

I will guarantee my reservation with: ☐ an advance deposit or ☐ credit card # _____

☐ American Express ☐ Visa ☐ MasterCard ☐ Diners ☐ Carte Blanche Expiration date _____

Please mail this form with your payment to: The Claremont Resort, P.O. Box 23363, Ashland & Domingo Avenues, Oakland, CA 94623, (415) 843-7926.

SYMPOSIUM COMMITTEE

Terry Vickers Benz
General Chair

Tom Berson
Program Co-Chair

Deborah Downs
Vice-Chair

Deborah Cooper
Program Co-Chair

**Please contact the IEEE Computer Society,
1730 Massachusetts Ave., N.W., Washington, DC 20036,
(202) 371-1013 for a copy of the complete
advance program.**

Sponsored by the **IEEE Technical Committee on Security and Privacy**
In cooperation with the **International Association of Cryptologic Research**

 **IEEE COMPUTER SOCIETY**

 **THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC.**
IEEE

Guest Editor's Introduction

Robotics and Automation

Gordon T. Wilfong, AT&T Bell Laboratories

Factory automation is widely recognized as an important element for remaining competitive in the manufacturing sector of the world economy. Advances in automation not only increase productivity and reduce costs but also increase the manufacturer's ability to quickly modify operations to produce state-of-the-art products.

The field of robotics has gained much from research in many fields, such as mechanical engineering, electrical engineering, applied physics, computer science. This issue of *Computer* focuses on the interaction between the theoretical foundations being developed in the computer science community and the practical considerations of those who wish to implement real-world systems.

Robotics research is concerned with the design and implementation of flexible, reliable automated systems capable of performing some class of actions while interacting with a physical environment. A sequence of robots performing various tasks along an assembly line or an autonomous vehicle exploring otherwise inaccessible areas are examples of such systems. Computer science researchers have responded to the needs of robotics, but many important areas require much further research.

To remain competitive in the modern manufacturing environment, manufacturers must rapidly respond to consumers' changing demands and quickly incorporate technological advances into their products. The earliest automated systems used hard automation. That is, they were

designed to produce one type of product, and any change in the product required shutting down the system while making and testing physical changes in the system. Thus, changes were not always economically feasible. In a flexible system, changes can be made quickly and simply by modifying the system's control software. These software changes are relatively inexpensive. Also, lengthy shutdowns for testing can be avoided through off-line tests using simulation. Thus, the problem of modeling and simulating objects subject to kinematic and dynamic constraints is an important and growing area of interest in the computer science community.

The need for flexible systems can be illustrated by considering autonomous vehicles that move parts between factory workstations. In most of today's systems, the vehicles move by following a wire embedded in the floor or a magnetic tape (or chemical trail) on the floor. Changing the pattern of the embedded wires is difficult and expensive because the floor must be ripped up. Material (such as tape or chemicals) on the factory floor provides a more flexible system, but it soon becomes worn and dirty and must be replaced. A truly flexible system would allow the vehicle to follow any computed path without external guides such as wires or tapes. The need for such systems has led computer scientists into the large and fruitful area called "motion planning." Given a description of the environment and the desired initial and final positions, a motion-planning algorithm computes a collision-free motion.

An automated system must be reliable

in the sense that it is guaranteed to behave and perform as specified. This requirement is important for safety in an environment where robots and other automated equipment could be hazardous to humans working nearby. Reliability is also relevant for economic reasons, since a batch of improperly manufactured product is likely to be an expensive error. Computer science research has been concerned with the development of provably correct algorithms and the provably correct software implementation of the algorithms. Many algorithms emphasizing proofs of correctness and efficient worst-case running time have been developed, but most are extremely complicated to implement.

Since an automated system moves among and manipulates physical objects, it must adjust its activities to conform to unexpected environmental features sensed during operation. For example, for safety reasons, an autonomous vehicle should sense objects in its path and avoid collisions. Since the system must deal with real-world objects, there is always uncertainty regarding the exact location and shape of the objects. This gives rise to interesting problems concerned with manipulating objects and avoiding collisions when complete information about the layout of the workspace is impossible. Also, since physical objects such as robots occasionally break down, systems must continue to perform reliably (or, at least, come to a safe halt) until a broken subsystem can be repaired. For instance, an autonomous vehicle might use an expensive vision system as its primary sensor, but on failure of that system, a less-accurate and less-



expensive acoustic ranger could be activated to allow the vehicle to reach a repair station without causing a collision. Similarly, if a robot drops a part it is carrying, it should sense the error and perform an error-recovery procedure, such as picking up the dropped part. An important implied requirement is that the system detect the failure of a subsystem. Similar problems have been widely studied in computer science in such areas as transaction management in databases and fault-tolerant distributed systems, but robotic recovery systems often have the additional requirement of recovery routines that are executable in real-time. Another difficulty is that potential errors frequently must be detected before they are executed. For example, once a too-large hole has been drilled in a piece of stock, it cannot be undone.

Another area of interest is the design of the product to be manufactured. Often, automated systems have been designed to manufacture a product that was designed to be built by humans. Not surprisingly, these systems are often less efficient than the humans they were expected to replace. Robots are superior to humans in some areas, such as the rapid insertion of chips on a board, and inferior in other areas, such as object recognition. So why should one expect a robot to outperform a human at a task designed with human abilities in mind? In designing an automated system to manufacture a particular product, the first step should be to design the product so that its manufacture is feasible given the robotics technology available. The problem of designing for ease of manufacture highlights many interesting questions for computer scientists concerned with such areas as geometric modeling, planning, and simulation.

The collection of articles in this issue presents a sample of the ongoing research in robotics. Each article gives a useful illustration of the various interactions between those individuals who are developing the theory and those who are concerned with actual

implementations. In the first article, "Algorithmic Motion Planning in Robotics," Micha Sharir surveys theoretical results in motion planning. The next article, "Task-Level Planning of Pick-and-Place Motion" by Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell, describes an ongoing project for implementing a system for planning a robot's grasps and motions. Christoph M. Hoffmann's "Problems of Accuracy and Robustness in Geometric Computation" discusses progress in developing geometric algorithms that do not require the assumption of exact calculations. Ingemar J. Cox and Narain H. Gehani's article, "Exception Handling in Robotics," offers a solution to the problem of dealing with errors that occur during robot operation. The issue concludes with surveys of robotics projects at four institutions. □

Acknowledgment

It is with great pleasure that I thank Varol Akman of Bilkent University, Ankara, Turkey, for his collaboration in the initial proposal for this special issue and for his assistance with some of the early administrative work.



Gordon T. Wilfong joined the Robotics Principles Research Department of AT&T Bell Laboratories, Murray Hill, New Jersey, in 1984. His current research interests include algorithmic motion planning and computational geometry.

He received the BSc degree in mathematics from Carleton University, Ottawa, Ontario, Canada, in 1980 and the MS and PhD degrees in computer science from Cornell University, Ithaca, New York, in 1983 and 1984, respectively.

Wilfong's address is AT&T Bell Laboratories, 600 Mountain Ave., Rm. 2C580, Murray Hill, NJ 07974-2070.



See the entire sequence by fanning and flipping pp. 7-49

About the cover

The cover photo strip is the first of three sequences that show a robotic system servicing a space vehicle. As simulated by Silma's CimStation system, two robot arms cooperate to accomplish the complex task of removing a payload from a closed enclosure.

In the first sequence, on the cover and on pp. 7-17 with intervening frames, the robotic system emerges from its housing and stretches out its arms. In the second sequence, facing page and pp. 19-35, one arm grasps a post on the space vehicle for stability while the second opens the enclosure's door and removes the payload. In the final sequence, below and pp. 37-49, the robotic system closes the door and readies itself for the next task.

The entire sequence simulates both kinematics and dynamics, including the physical forces acting on the robotic system and its ability to succeed with tasks despite these forces. NASA plans to use robotics extensively on spacecraft to reduce the danger to astronauts and the expense of manned voyages.

CimStation graphics software for designing, programming, and simulating automated workcells is a product of Silma, Inc., 1601 Saratoga-Sunnyvale Rd., Cupertino, CA, 95014, phone (408) 725-8908.



International: Please call
0101-617-491-7311 for your
nearest dealer.

If you program in C,
wouldn't you like to save 40%-80%
of development time?

And enjoy doing it?

Use the advanced C-scape™ Interface
Management System to create your screens, and
most of your work will be done before you start.
Best of all, it's fun.

Programmers in over 30 countries are saving time and enjoying using C-scape. Their screens are elegant. Their windows are dynamic. Their menus are fast. Their data entry is validated. Their help is cross-referenced. Their words wrap in text fields. And their users are delighted.

Airlines, engineering firms, oil companies, medical instrument companies, electronics companies, hospitals, automobile manufacturers, universities, and international financial institutions are using C-scape to produce commercial, industrial, scientific, technical, and in-house applications in far less development time.

Specifically, C-scape users report saving an average of 40% of overall development time. Some users report savings beyond 80%. They also say that programming is actually more enjoyable using C-scape.

IEEE software reviewers say using C-scape is "a pure joy, primarily due to the ease of use and the extensive set of high-level functions included in the package." The reviewers report, "We have used a number of different screen management systems in the past, but C-scape is by far the best we've encountered to date." (IEEE COMPUTER, February 1988)

Why is C-scape used in such diverse but critical applications? Because C-scape gives you the power, flexibility, leverage, and dependability you need to create, maintain, and port even the most difficult applications.

The new C-scape 3.0 incorporates three years of improvements.

- The function library includes fully documented routines for windows, menus, data entry and validation, word-wrap text editing, context-sensitive help. An array of other functions—time, money, keyboard playback, and more—saves you *even more time*. C-scape works in text and graphics modes, so you can combine C-scape with graphics screens.
- The all new Look & Feel™ Screen Designer lets you create screens and forms. Then it automatically turns them into C code or binary files callable at run time. Look & Feel also turns Dan Bricklin Demo (or any ASCII) screens to C, saving time and reducing error.
- The function-building toolkit makes it easy to create your own functions. Develop your own look and feel for your function library. These unique toolkit functions are key to modifying existing functions and creating new ones.
- A device interface group contains all the hardware dependent functions making C-scape code portable virtually anywhere. Video RAM, ROM BIOS, ANSI support, mouse support, automatic CGA, EGA, VGA (including 43 line mode) support. And they're swappable at run time.

Using the package is easy. Call the appropriate functions from your code. Or, let Look & Feel produce the code for you. Control and modify anything you like. C-scape is unsurpassed in flexibility.

Best of all, C-scape is fun and extremely easy to learn. If you know printf, you can use C-scape. C-scape runs under MS-DOS, OS/2, UNIX, XENIX, VMS, Apollo, Sun, and others.

To get started, order C-scape on a 30-day review basis. You'll get a professional, fully-indexed manual with numerous source code examples and a complete function reference. A demo disk provides working code examples. The famous Look & Feel Screen Designer will have you producing compilable source code in minutes.

We offer international technical support and a 24-hour bulletin board. You get complete library source code at no additional cost. We automatically notify you of upgrades and other work-saving Oakland products. C-scape is fully compatible with the C-cell™ Spreadsheet System and other Oakland C tools.

Please call for technical literature, a demo package that includes Look & Feel, or better yet, place an order. Call now. You'll save time. And be pleased.

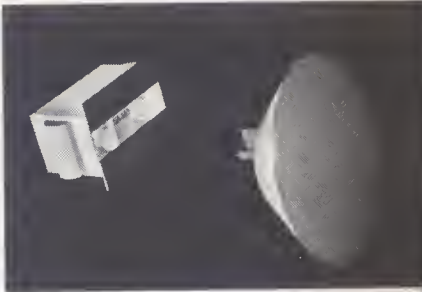
800-233-3733 (In Massachusetts, 617-491-7311) MasterCard and VISA accepted.

DOS and OS/2: Library only (includes source code) \$299. Look & Feel Screen Designer \$149. Both (Library and Screen Designer) \$399. UNIX, XENIX, VMS, Apollo, Sun, Stratus, others, please call. We do not charge royalties or run time licenses. FAX: 617-868-4440.

OAKLAND C TOOLS

Oakland Group, Inc. • 675 Massachusetts Avenue • Cambridge, Massachusetts 02139-3309, U.S.A. • Prices and terms subject to change.

Reader Service Number 1



Algorithmic Motion Planning in Robotics

Micha Sharir

New York University and Tel Aviv University

Future generations of robots will be considerably more autonomous than present robotic systems. The main objective of research on theoretical problems in robotics is to endow robotic systems with basic capabilities they will need to operate in an intelligent and autonomous manner. These improved capabilities fall into three broad categories:

- *Sensing*—The system should be able to gather information about its workspace through a variety of sensing devices (visual, tactile, or proximity sensing) and analyze and transform the raw sensory data into a world model of the environment.

- *Planning*—This model is then used to plan tasks the robot is commanded to execute, such as navigation, assembly, inspection, and machining.

- *Control*—The system should finally obtain a low-level control loop that monitors the actual execution of each planned substep of the task.

The aim at the planning stage is to allow the robot's user to specify a desired activity in very high level, general terms and have the system fill in the missing low-level details. For example, the user might specify the end product of an assembly process and ask the system to construct a



An intelligent tool, or a robot that thrashes about uselessly in an environment it cannot apprehend? The difference may lie in the painstaking development of geometric algorithms.

sequence of assembly substeps; or, at a less demanding level, to plan collision-free motions that pick up individual subparts of an object to be assembled, transport them to their assembly position, and insert them into their proper places.

Studies have shown planning to have significant mathematical content. Tools drawn from various subfields of mathematics—classical geometry, topology, algebraic geometry, algebra, and

combinatorics—have all proved relevant. The algorithmic approach to robot planning, emphasized in this article, uses a variety of tools drawn from computational geometry and other areas of computer science. It involves the design and analysis of geometric algorithms, which are required to be nonheuristic, exact (if the input data is accurate), and asymptotically efficient in the worst case. The motion-planning problem has been studied intensively over the last decade and has been found to possess a substantial amount of mathematical depth and sophistication, as the reader will probably realize from this article.

In addition, the interaction between motion planning and computational and discrete geometry has produced rich cross-fertilization between the fields, leading to many deep new results in geometry as well as to better understanding of the motion-planning problem. Some of the efficient algorithms devised have already been implemented successfully.

The approach to motion planning surveyed here is relatively new. It emphasizes object-oriented, exact, and discrete (or combinatorial) algorithmic techniques in which worst-case asymptotically efficient solutions are being sought. It has been

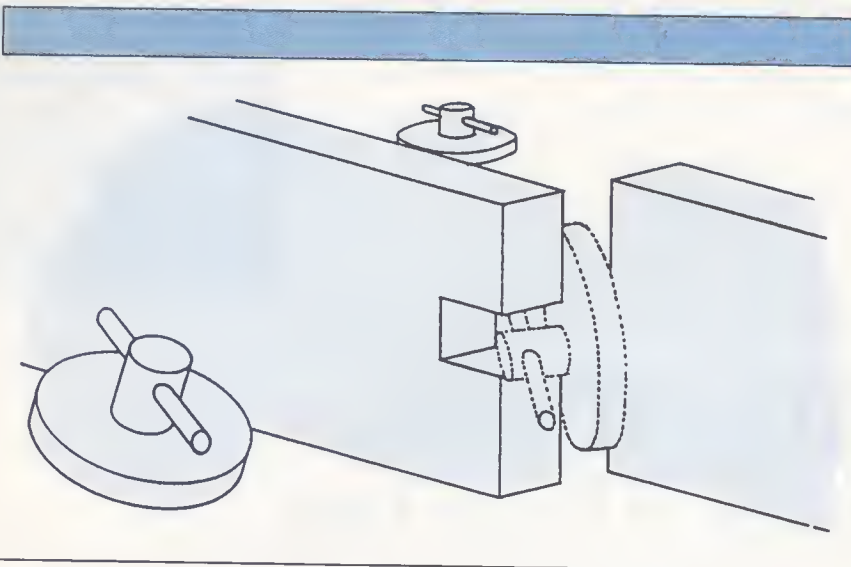


Figure 1. An illustration of the motion-planning problem.

preceded by (and currently “competes” with) several alternative approaches that are usually more engineering or AI based and that emphasize heuristic, approximating, rule-based, or best-case-tailored solutions. Several interesting solutions of this kind have been developed by Lozano-Perez, Brooks, Mason, and Taylor.¹ Lozano-Perez provides an example.¹ Other approaches are nondiscrete in that they view the motion-planning problem as an integral part of low-level robot control. Such an approach typically regards obstacles as sources of repelling potential field, while the target placement of the robot is regarded as a strong attractor. The hope is that the moving body, as it attempts to follow the potential gradient vector field, will be able to reach the destination and avoid colliding with obstacles. Yet another approach, nonheuristic but not quite discrete, has been developed by Lumelsky and his collaborators.² It applies when the environment is not known a priori.

We can bypass a discussion of which method is superior, since each has advantages and disadvantages. Many of the heuristic approaches have been implemented for several years now and have working prototypical systems capable of planning motions in a variety of situations. However, because the problem has such a rich geometric and combinatorial structure, it is imperative that we understand this structure in detail from a mathematically rigorous point of view and explore it for potential algorithmic improvements before beginning to negotiate approximating or heuristic shortcuts. Such shortcuts,

if taken naively, can completely miss this structure, often resulting in considerable loss of efficiency at best, and, at worst, in a complete misunderstanding of the issues at the heart of the problem. As we will see below, exact solutions of the problem for complex moving systems might be (provably) computationally intractable in the worst case, so, in practice, heuristic shortcuts might be necessary. Nevertheless, they can be made most effectively once the exact solution and its limitations are well understood.

Statement of the problem

In its simplest form, the motion-planning problem can be defined as follows: Let B be a robot system comprising a collection of rigid subparts, some of which might be attached to each other at certain joints, while others might move independently. Suppose B has a total of k degrees of freedom, that is, each placement of B can be specified by k real parameters, each representing some relationship (orientation, displacement, etc.) between certain subparts of B . Suppose further that B is free to move in a two- or three-dimensional space V amidst a collection of obstacles whose geometry is known to the robot system. Typical values of k range from 2 (for a rigid object translating on a planar floor—without rotating—for example) to 6 (the typical number of joints for a manipulator arm). The values can also be much larger—for example, when

we need to coordinate the motion of several independent systems in the same workspace.

The motion-planning problem for B is

Given an initial placement Z_1 and a desired target placement Z_2 of B , determine whether there exists a continuous obstacle-avoiding motion of B from Z_1 to Z_2 , and, if so, plan such a motion.

Figure 1 illustrates this problem.

Since we are interested in an exact, nonheuristic, and nonapproximating algorithmic approach, we want any algorithm for solving the motion-planning problem always to be able to find a motion, if one exists. It is intuitively clear that in favorable layouts of the obstacles (for example, relatively few obstacles, well separated from each other), even simple-minded techniques will generally yield a solution. However, if the obstacles are densely cluttered, a solution might be quite difficult—often beyond normal human capabilities. In such circumstances the exact algorithmic approach has the highest payoff.

Many useful extensions and variants of the problem can arise in practice. For example, if the environment is not fully known to the system, it must use its sensors continually to update the environment model and its planned motion. Or perhaps the objects aren't stationary but rather move along predictable and known trajectories, in which case the system needs to “dodge” the obstacles while moving toward its goal. Yet another case is when the geometric model of the environment is not accurately known. The system will then need to allow for possible errors and come up with a robust planning strategy that would adjust itself automatically when unanticipated contacts are made. This is crucial in cases of compliant motion, where the system moves in contact with some obstacle whose shape is known only approximately.

Other variants require the planned motion to be optimal in the sense that it minimizes the time needed for its execution, the total distance traveled, or the total energy consumed. Also, the problem has been posed in such a way that it ignores the dynamics of the moving system. For example, the system might be constrained so that its velocity and/or acceleration cannot exceed certain thresholds, in which case even simple motions such as turning a corner might be nontrivial to plan.

We will consider these extensions of the problem later, but for now let's focus on the simpler, static, and purely geometric

version of the problem as posed above. It is difficult enough to solve the problem as it is, and each of the extensions just mentioned makes the solution more complicated.

We can obtain an alternative formulation of the problem, which will help us understand its geometric context, as follows: Let FP denote the "free configuration space" of the moving system B ; that is, FP is the k -dimensional parametric space of all free placements of B (the set of placements of B in which B does not intersect any obstacle). Each point Z in FP is a k -tuple giving the values of the parameters controlling the k degrees of freedom of B at the corresponding placement. For example, if B is a planar rigid body moving in a planar region, then it has three degrees of freedom, and each placement of B can be specified by three parameters (x, y, θ) , where (x, y) are the coordinates of some fixed reference point on B , and θ is the angle between some fixed reference axis attached to B and the x axis. As another example, if B is a line segment moving in three-dimensional space, B has five degrees of freedom and each placement of B can be specified by a 5-tuple giving the x, y , and z coordinates of one designated endpoint p of B and the spherical coordinates of the direction along B from p to the other endpoint.

Obviously, not every placement of a moving system B is free. For a placement Z to be free, it must satisfy a collection of *collision constraints*, each of which forbids a specific feature of B (a corner, a straight link, etc.) to contact or to intersect a specific feature of some obstacle (a vertex, an edge, a face, etc.). Each such collision constraint corresponds to a "forbidden region" in the configuration space, and the free configuration space is simply the complement of the union of these regions.

In this setting the motion-planning problem becomes a problem in "computational topology," that is, a problem of calculating and obtaining a discrete representation of FP . More precisely, one needs to decompose FP into its path-connected components, because a collision-free path exists between the two given placements Z_1, Z_2 of B if and only if they both lie in the same connected component of FP . An appropriate discrete representation of the connected components of FP usually assumes the form of a "connectivity graph" whose nodes represent certain local portions of FP and whose edges connect portions adjacent to each other in FP . Once such a representation

has been obtained, actual motion planning reduces to searching for a path in the connectivity graph that connects the initial and final placements of B . Having found such a path, the system finally transforms it into a sequence of actual elementary motions, each of which can be executed by a single command to the motors controlling the robot motion.

The next section discusses the inherent computational complexity of the basic motion-planning problem, details several general techniques developed for treating it, and reviews various efficient algorithms for planning the motions of certain special robot systems having only a few degrees of freedom.

In more detail, I will show that when the number k of degrees of freedom becomes arbitrarily large, the problem complexity increases substantially. For certain types of systems with an arbitrarily large value of k , such as a collection of many independent rectangular objects moving in a coordinated fashion inside a rectangular region, the problem becomes computationally intractable (specifically, $Pspace$ -hard), and it isn't likely that efficient worst-case solutions will ever be found. Intuitively, this is because when k is large, the space FP is a high-dimensional space with irregular boundaries, making it hard to calculate efficiently. General techniques have been developed for solving the motion-planning problem in full generality, and this article describes some of them. But, as just indicated, these general algorithms are hopelessly inefficient in the worst case, and although their development is significant from a theoretical point of view, they are completely useless in practice.

However, when k is small, the problem becomes more tractable, and efficient algorithms have been developed for a variety of simple systems. A prototypical example has been that of a line segment (a "ladder," a "rod," or a "pipe") moving in the plane amidst polygonal obstacles,³⁻⁶ but many other systems have also been handled successfully. The next section describes these specialized algorithms.

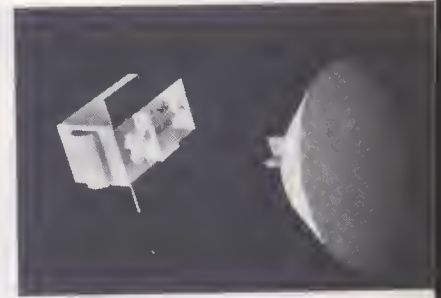
The last section of this article looks at some of the extensions to the motion-planning problem mentioned above, including cases of moving obstacles, optimal motion, and an unknown or only partially known environment. Of course, these extensions are substantially more difficult to handle than the original problem version, making efficient solutions much

harder to obtain, except in certain very special cases.

This difficulty becomes apparent, for example, when we try to produce a collision-free path that satisfies some criterion of optimality. If a mobile robot is approximated as a single moving point, we might want to find the Euclidean shortest path between the initial and final system placements that avoids (the interiors of) the given obstacles. In more complex situations the notion of optimal motion is less clearly defined and has hardly been studied. However, even in the simplified case of the Euclidean shortest path problem, the situation becomes more complicated. While in the planar case the shortest path can be found in time that is quadratic in the number of obstacle corners and edges, finding the shortest path between two points that avoids a collection of polyhedral obstacles in three dimensions is already computationally intractable. In contrast, the coarser original problem of simply determining whether a path exists at all can be solved by a reasonably efficient algorithm, even in three dimensions.

Motion planning in static and known environments

The free configuration space FP of the moving system B is determined by the collection of equalities and inequalities expressing the fact that at a free placement the system B avoids collision with any of the obstacles present in its workspace. I will denote by n the number of equalities and inequalities needed to express all these collision constraints and call it the "geometric (or combinatorial) complexity" of the given instance of the motion-planning problem. We will reasonably assume that



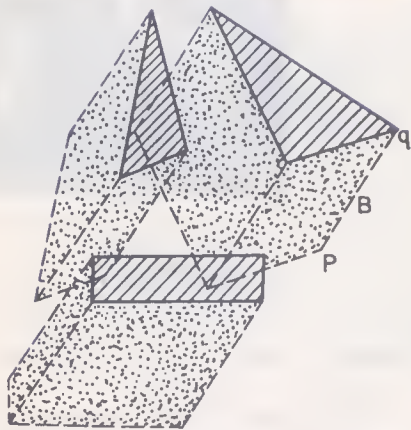


Figure 2. Constraint curves and the space FP for a translating segment.

the parameters describing the degrees of freedom of B can be chosen in such a way that each of these inequalities is algebraic. Indeed, the group of motions (involving various combinations of translations and rotations) available to a given robot can ordinarily be given algebraic representation, and the system B and its environment V typically can be modeled as objects bounded by a collection of algebraic surfaces (for example, polyhedral, quadratic, or spline-based).

To illustrate these concepts and get a somewhat better feeling for the structure of FP , consider the following simple example. Let B be a line segment pq translating in the plane amidst a collection of polygonal obstacles having a total of n corners (or edges). Since B is not allowed to rotate, its position is fully specified by the (x, y) coordinates of its endpoint p . The forbidden regions of collision are bounded by a collection of *constraint curves*, each expressing a possible contact between some fixed feature of B and some fixed obstacle feature. In this case each such contact is between an endpoint p or q of B and some obstacle edge, or between an obstacle corner and the line segment B . A fairly simple geometry exercise will show that each of these curves (namely, the locus of the endpoint p of B as it makes such a contact) is a line segment either at the orientation of B (in case of contact

between B and an obstacle corner) or otherwise parallel to the corresponding obstacle edge. Thus the forbidden regions are each polygonal, so that their union (and thus its complement FP) is also a polygonal region bounded by portions of these constraint curves (see Figure 2). Note that the number m of constraint curves is proportional to n .

Another way of thinking about FP is to consider the *arrangement* of the m constraint segments, that is, the planar subdivision obtained by drawing all these segments. Each face of this arrangement (a connected component of the complement of the union of the segments) is either fully contained in FP or fully contained in a forbidden region. This follows easily from the fact that any transition between a free region and a forbidden one must occur at a point lying on some constraint segment. Thus FP is simply a collection of faces of the constraint segments arrangement. These arguments lead to a crucial observation that simplifies the problem still further. That is, if the initial placement Z_1 of B is known in advance, then it suffices to calculate only the connected component (face) of FP that contains Z_1 , because no other free placement can ever be reached from Z_1 by a continuous obstacle-avoiding motion.

In this setting the special case of the motion-planning problem just discussed has been reduced to the problem of calculating a single connected component in an arrangement of m segments in the plane. Essentially the same arguments show that to solve the motion-planning problem for an arbitrary system with two degrees of freedom, it suffices to calculate a single face in an arrangement of arcs in the plane. When the system has more degrees of freedom, the problem becomes that of calculating a single cell in an arrangement of (algebraic) surface patches in a higher dimensional space.

To recap, a typical instance of the motion-planning problem will involve a collection of n constraint surfaces in the k -dimensional parametric space of all system placements, where each surface is a patch of some algebraic surface, and our goal is to analyze and calculate a single cell in the arrangement of these surfaces. In most practical instances the maximal degree of these algebraic surfaces is fixed and small (because these surfaces describe only a few types of possible contacts between B and the obstacles), while the number of surfaces is generally large. Thus what follows

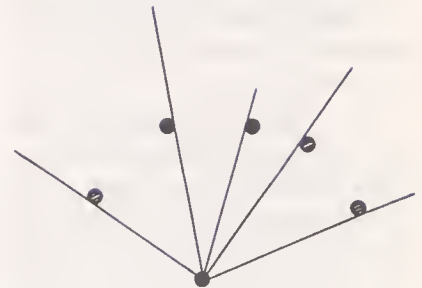


Figure 3. An FP vertex for a spiderlike system.

tends to ignore the algebraic complexity of handling these surfaces (for example, finding the intersection points of any k of these surfaces) and concentrates on the combinatorial complexity of their arrangement, that is, the size of a discrete representation of the arrangement as a function of the number of surfaces n . Nevertheless, the algebraic complexity of the motion-planning problem is also an important research area.

To get a feel for the structure of such an arrangement, think of a simple special case involving flat triangular surfaces in three-dimensional space. (This special case actually arises in the motion-planning problem for a polyhedral object B translating in three-dimensional space amidst polyhedral obstacles; see below.) These triangles decompose three-dimensional space into connected cells, the boundary of which consists of two-dimensional portions of these triangles that meet along one-dimensional edges, each being the intersection segment of a pair of triangles. These edges terminate at endpoints, where each endpoint is either an intersection point of three triangles or an intersection of one triangle's edge with the interior of another. If we think of each triangle as the locus of placements of B in which it makes some specific contact with some obstacle, then edges of the arrangement (that appear along the boundary of FP) correspond to

loci where B makes two different obstacle contacts simultaneously (while otherwise remaining free of collisions). In addition, vertices correspond to placements of B at which it makes three simultaneous contacts with the obstacles. Abstracting to a general instance involving k degrees of freedom, we see that each face on the boundary of FP consists of placements of B in which it makes some $j \leq k$ distinct contacts with the obstacles simultaneously, while otherwise remaining free. A simple combinatorial argument implies that the number of faces constituting the boundary of FP is at most proportional to n^k . Specifically, consider the vertices of FP . Each vertex is an intersection point of k of the n surfaces. There are $\binom{n}{k}$ distinct k -tuples of the given surfaces, and, by Bezout's classical theorem in algebraic geometry, each such k -tuple of surfaces intersects in at most d^k points, where d is the maximum degree of the surfaces. Assuming d is small, thus ignoring the factor d^k , the claim follows.

To see that the complexity of FP can be that high, consider a simple spiderlike planar system B consisting of k rigid straight links all joined at a common endpoint that is constrained to lie at a fixed point O in the plane, so that each link can rotate independently around O . This system has k (rotational) degrees of freedom. Suppose the obstacles consist of n small disks arranged along some circular arc about O (see Figure 3). Thinking of FP as a semialgebraic set in E^k , a vertex of its boundary corresponds to a placement of B where k obstacle contacts are being made simultaneously, that is, where each link touches some obstacle. Since we can choose the k contacts out of the n obstacles, it follows that the boundary of FP can have about n^k vertices, as claimed.

At this point the following observations are prudent. Even though the combinatorial complexity of the entire FP might be high, it is conceivable that to plan a motion of B , one does not need to calculate FP in full. Indeed, we have already observed that it suffices to calculate just a single connected component of FP . But even that might be too much when all we really want to produce is a collision-free path, or, in a further-restricted version of the problem, just to determine whether such a path exists. To bring this argument to the extreme, suppose FP is known to be a convex region. Then we do not need to compute it at all, because any two placements in FP can be connected by the

straight segment joining them, which necessarily lies within FP . We will return to this issue later, but to start, let's first consider techniques that do aim to calculate the entire FP .

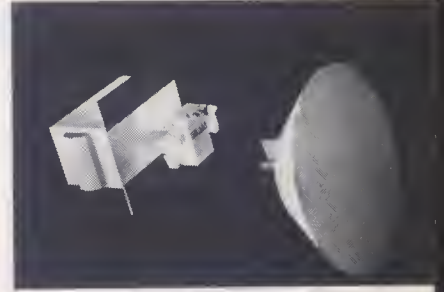
General solutions to the motion-planning problem. Assuming then that FP is an algebraic or semialgebraic set in E^k , it has been shown⁷ that the motion-planning problem can be solved in time polynomial in the number n of (algebraic) geometric constraints defining FP and in their maximal degree, but doubly exponential in k . This initial result has since been extended and improved in several papers, culminating in the recent fundamental work by Canny.⁸ Canny provides a general technique for solving the motion-planning problem for a system with k degrees of freedom under n geometric constraints of fixed maximal degree, in time $O(n^k \log n)$,* where unfortunately the constant of proportionality, which depends on the maximal degree of the collision constraints, is prohibitively large.

Still, from a theoretical point of view, this has been significant progress, because, as noted above, the combinatorial complexity of FP can indeed be as high as proportional to n^k in the worst case, which makes Canny's algorithm nearly worst-case optimal. Canny and Renegar later improved the technique slightly by showing that it can be implemented in P space (that is, using only a polynomial amount of storage).

Both of the above-mentioned techniques are fairly complicated and apply to general semialgebraic sets to produce some discrete representation of such a set's topological structure. The following is only a sketchy description of these techniques.

Let's begin with the technique Schwartz and I developed. Although it gives inferior performance, this method is significant because of its (relatively) simple structure, which makes variants of the method quite attractive for solving the motion-planning problem for several special instances. This technique is a variant of Collins' cylindrical algebraic decomposition technique, which was originally applied to Tarski's theory of real closed fields. (See our article for a review of Collins' technique.⁷) In Tarski's theory one is given a quantified

*I assume the reader's familiarity with the "big-Oh" notation for asymptotic combinatorial and computational complexity bounds.



Boolean formula involving k variables that can assume real values and containing atomic subformulas that are equalities and inequalities involving polynomials in these k variables. The problem is to decide whether the formula is true.

Collins' technique solves this problem by decomposing the k -dimensional space into a finite number of connected cells of various dimensions, such that over each cell each of the polynomials appearing in the formula assumes a constant sign (zero, positive, or negative). Let P denote the set of polynomials appearing in the formula. The cells are constructed in an inductive manner. In one dimension they are merely a decomposition of the line into open intervals and their endpoints, which are the real zeros of the polynomials in P . In $k+1$ dimensions we first obtain a cylindrical decomposition of the k -dimensional space of the first k coordinates, for a different set Q of k -variate polynomials, derived from the original set P . Then, for each "base-cell" c , we decompose the cylinder $c \times R$ over c by cutting it into layers at its intersections with the zero-surfaces of the polynomials in P .

Collins' original work shows how to use these cells to decide the truth of the given formula by essentially testing the formula at a single point in each cell. It also shows how to compute effectively all these cells—using only purely rational arithmetic—by representing each cell by an algebraic point inside it (given implicitly so that its coordinates are roots of certain specified polynomials), plus some other tricks. The number of cells, as well as the time needed to produce them, is shown to be doubly exponential in the number of dimensions k but to depend only polynomially on the number of polynomials and on their maximal degree.

In terms of the motion-planning application, if we apply the Collins decompo-

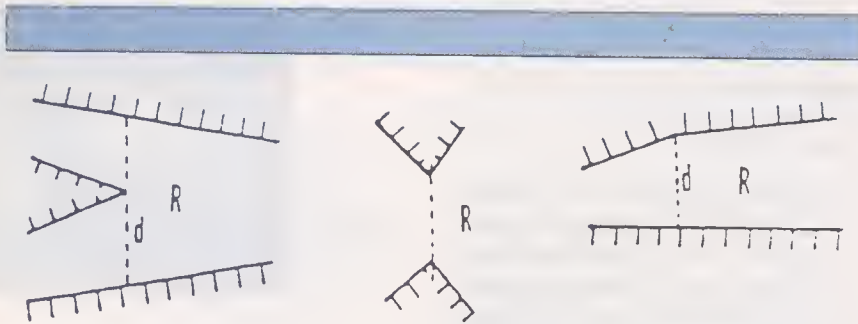


Figure 4. Critical orientations for a moving rod in the plane.

sition technique to the collection of collision constraints, each cell c in the resulting decomposition has the property that each constraint has a fixed sign over c . Consequently, c is either fully contained within FP or lies fully in the forbidden region. Thus by scanning each of the Collins cells, we can collect all cells whose union forms FP .

This is still not quite sufficient to solve the motion-planning problem, because we also need to know how these cells are connected ("glued") to each other in FP so that we can deduce the topological structure of FP . Schwartz and I supply this extra component and give a somewhat complicated method for determining all pairs of adjacent Collins cells in FP .⁷ Subsequent improvements to this method have recently been provided.

The motion-planning problem is then solved by constructing a connectivity graph CG whose nodes are the Collins cells constituting FP and whose edges connect pairs of adjacent Collins cells. Then, for a given pair of initial and final placements Z_1, Z_2 of B , we locate the two nodes c_1, c_2 of CG that contain these placements and then search for a path in CG connecting c_1 and c_2 . If such a path exists, B can be moved continuously from Z_1 to Z_2 without collision along a path easily constructed from the graph path; otherwise, no such motion is possible.

Many components of the above technique (decomposing FP into simple cells, establishing adjacency between the cells, and transforming the problem to a discrete path-searching in a graph) have proved useful in many related approaches, where

efficient solutions for specific systems have been obtained (see below). However, the general method as sketched above is very inefficient because it produces too many cells, most of which appear not to be required. Still, no sufficiently general improvement of the method is known.

Canny's technique is quite different and has a significantly smaller computational complexity. Instead of decomposing FP into cells, he suggests constructing a one-dimensional skeleton S (which he calls a "road map") of curves within FP . This skeleton has two crucial properties:

- **Reachability**—Each placement Z in FP can effectively be moved continuously to a placement in S .
- **Connectivity**—The intersection of S with each connected component of FP is nonempty and connected.

Clearly, if such a road map can be found, the motion-planning problem is easily solved as follows: Apply the reachability procedure to the initial and final placements Z_1, Z_2 of B to obtain two respective placements W_1, W_2 on S . Then search for a path in S (which is easy to do, since S is one dimensional) between these two placements. If no path exists, then the connectivity property of S guarantees that Z_1 and Z_2 cannot be reached from one another by a collision-free continuous motion. Otherwise, the required motion is simply the concatenation of the paths connecting Z_1 to W_1 and Z_2 to W_2 and the path connecting W_1 to W_2 within S .

To obtain a road map, Canny uses a recursive procedure on the dimension k of FP . If $k = 1$, then FP is a curve and is taken

to be its own road map. Otherwise, FP is projected onto some two-dimensional plane, and the *silhouette* (boundary curve) Σ of the projected image is formed. Next, one finds all critical levels of Σ , that is, the y coordinates (in the projected plane) of points at which Σ is either singular or has a horizontal tangent. Then, FP is sliced at each of these critical levels, and the algorithm is applied recursively to each cross section. Finally, all road maps obtained for each cross section are glued to the silhouette Σ , and the union of all these curves and of Σ is the desired road map for FP . Canny⁸ shows that this road map satisfies the *reachability* and *connectivity* conditions described above.

This basic algorithm is not as efficient as possible because the slices of FP through successive critical levels of the silhouette are essentially very similar in shape, except for a "local" change that occurs near the points of criticality of Σ . Using this observation, Canny derives an improved algorithm whose running time is shown to be

$$O(n^k(\log n \cdot d^{O(k)} + (dk)^{O(k^2)}))$$

where d is the maximum algebraic degree of the constraint polynomials (and where we have omitted factors that depend on the magnitude of the coefficients of the constraint polynomials). In other words, if d and k are both small and fixed, the time complexity of Canny's algorithm is $O(n^k \log n)$.

Lower bounds. The results just cited suggest that motion planning rapidly becomes harder as the number k of degrees of freedom increases; this conjecture has in fact been proved for various model robot systems. Specifically, Reif proved that motion planning is *Pspace*-hard for a certain 3D system involving arbitrarily many links and moving through a complex system of narrow tunnels.⁹ (A computational problem is *Pspace*-hard if it is at least as difficult to solve as any problem that can be solved using storage whose size is a polynomial function of the input size, making it highly unlikely that it will ever be solvable by a worst-case efficient algorithm.)

Since Reif's result, *Pspace*-hardness has been established for simpler moving systems, including 2D systems of mechanical linkages¹⁰ and a system of 2D independent rectangular blocks sliding inside a rectangular box.¹¹ Several weaker results establishing NP-hardness (a somewhat weaker form of intractability) for still sim-

pler systems have also been obtained.

The preceding results are interesting because they provide two relatively simple physical devices for simulating bounded-tape Turing machines. In addition, these lower bounds are fairly strong, because they apply to the decision problem in motion planning (which only has to determine whether two given placements are reachable from one another) and do not presume that the entire configuration space, or even the path itself, needs to be constructed.

A few lower bounds for simple systems with just a few degrees of freedom deserve mention here. Ke and O'Rourke⁶ consider the problem of a line segment moving in a 2D polygonal space with n corners and have shown that in the worst case there exist two placements of the rod that are reachable from one another, but any motion between these placements requires at least a number of elementary submotions proportional to n^2 (the rod must be pushed forward and backward at least that many times to reach its destination). Ke and O'Rourke also obtain a similar lower bound for a rod moving in three-dimensional space amidst polyhedral obstacles with n vertices, where the minimum number of submotions needed to realize a motion of the rod between two placements can be in the worst case as high as proportional to n^4 .

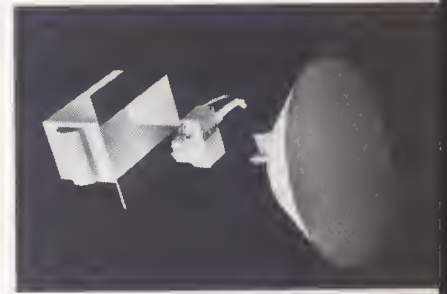
The projection method. The negative worst-case results mentioned above suggest that when the number k of degrees of freedom is large, the problem becomes intractable. (Of course, this does not make attacking such general problems hopeless. As with many other intractable problems, there might still be algorithms that work very well in the average case, or approximating algorithms that can guarantee, for example, that they will find a path efficiently, provided the "clearance" between obstacles is not too small. However, I will not address these issues here.) When k is small, Canny's and Schwartz and Sharir's general algorithms show that the problem can be solved efficiently (in time polynomial in the number n of constraints). Still, even in such special cases the efficiency of these general techniques is far from satisfactory. Consequently, much recent research has focused on obtaining improved efficient algorithms for a variety of special systems having just a few degrees of freedom.

We can classify these algorithms

broadly according to several general approaches they exemplify. The first approach, known as the projection method, uses ideas similar to those appearing in the Collins decomposition procedure described above. First we fix some of the problem's degrees of freedom (for the sake of exposition, suppose just one parameter y is fixed, and let $\bar{x}$ be the remaining parameters); then we solve the resulting restricted $k-1$ -dimensional motion-planning problem. This subproblem solution must be such that it yields a discrete combinatorial representation of the restricted free configuration space (essentially, a cross section of the entire space FP) that does not depend continuously on the final parameter y and changes only at a finite collection of "critical" values of y . These critical values are then calculated; they partition the entire space FP into connected cells. By calculating relationships of adjacency between these cells, we can describe the connectivity of FP by a discrete connectivity graph CG in much the same way as in the Schwartz and Sharir general algorithm.

Schwartz and I applied this relatively straightforward technique in a series of papers on the "piano movers" problem to yield polynomial-time motion-planning algorithms for various specific systems, including a rigid polygonal object moving in 2D polygonal space,³ two or three independent disks moving in coordinated fashion in 2D polygonal space, certain types of multi-arm linkages moving in 2D polygonal space,¹² and a rod moving in 3D polyhedral space.¹³ These initial solutions were coarse and not very efficient; subsequent refinements have improved their efficiency substantially.

A typical and extensively studied example is that of a line segment B (a "rod") moving in 2D polygonal space with a boundary consisting of n segments ("walls"). Here the configuration space FP is three dimensional, and, using a modified projection technique that Leven and I developed,⁵ it can be decomposed efficiently into cells. Briefly speaking, this technique varies the orientation θ of B continuously and stops at critical orientations, in which the combinatorial structure of the θ cross section of FP changes discontinuously (see Figure 4). Leven and I show that the number of critical orientations is at most $O(n^2)$, and that, assuming B and the walls are in "general position," each critical orientation delimits only a small constant number of cells in a certain cell



decomposition of FP . Thus the total number of cells in FP is also $O(n^2)$. We present a fairly straightforward algorithm, which runs in time $O(n^2 \log n)$, for constructing these cells and for establishing their adjacency in FP , a very substantial improvement of the $O(n^5)$ algorithm originally presented earlier.³ Moreover, in view of Ke and O'Rourke's lower bound mentioned above,⁶ the Leven-Sharir algorithm is nearly optimal in the worst case.

The retraction method. Several other important algorithmic motion-planning techniques were developed subsequent to the simple projection technique originally considered. The so-called retraction method is very similar to (and in fact preceded) Canny's general technique, described earlier. Specifically, it proceeds by retracting the configuration space FP onto a lower dimensional (usually a one-dimensional) subspace N , which satisfies both the reachability and the connectivity properties mentioned earlier, thus ensuring that two system placements in FP lie in the same connected component of FP if and only if their retractions to N lie in the same connected component of N . This reduces the dimension of the problem, and if N is one dimensional, the problem again becomes one of searching a graph. (Rigorously speaking, a retraction is a continuous map from some space E onto a subspace F , which becomes the identity when restricted to F . The retractions used in this approach are not always everywhere continuous, but their discontinuities are easy to characterize and to "patch up.")

O'Dunlaing and Yap¹⁴ have used this retraction technique in the simple case of a disk moving in 2D polygonal space. In this case each placement of the disk D can be fully specified by the position of its center in the plane. The retraction that is used attempts to repel D away from its nearest

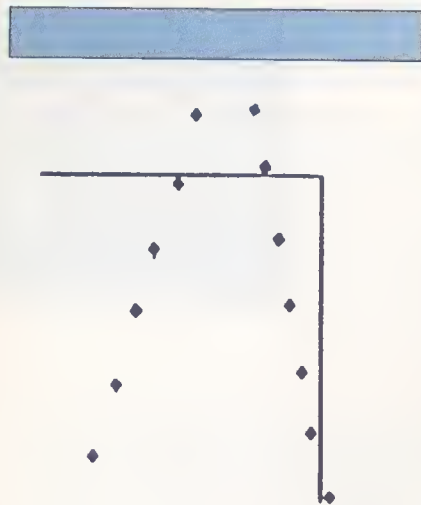


Figure 5. Quadratic-size *FP* for a translating nonconvex planar object.

obstacle, until its center becomes equally near to at least two distinct obstacle features. The locus N of such points is known as the *Voronoi diagram* associated with the set of given polygonal obstacles. It was recently shown that the Voronoi diagram of n line segments in the plane can be calculated in time $O(n \log n)$. After computing the diagram and pruning away portions of it where the nearest obstacle is too close for D to fit, we obtain the desired network N in total time $O(n \log n)$.

There have been several extensions of this idea to more complex systems, such as a moving line segment or a translating convex polygon, in the plane. Canny's results also follow the idea of constructing a one-dimensional skeleton network in *FP* that is large enough to represent the connectivity of *FP*.

Other techniques. A rather early paper by Lozano-Perez and Wesley¹⁵ introduced a general approach, known as the *expanded obstacles* approach, to a variety of motion-planning problems. This idea can be illustrated by the simple case of a convex polygon B , with k sides, translating in a 2D polygonal space. (Subsequently, an alternative retraction-based technique was developed for this case.) Suppose the obstacles are decomposed into m convex pieces (with pairwise disjoint interiors) that have n corners overall. Let B_0 be a standard placement of B in

which some fixed reference point O within B lies at the origin. For each convex obstacle A_i , consider the *expanded obstacle* A_i^* , which is defined as the Minkowski (or vector) difference of A_i and B_0 ; that is,

$$A_i^* = \{x - y : x \in A_i, y \in B_0\}$$

where $x - y$ denotes vector difference. If B is moved so that O lies at some point z of A_i^* , then by definition $z = x - y$ for some x in A_i and y in B_0 , so $z + y$, which by definition is a point occupied by the current placement of B , is equal to x , which is a point in A_i . In other words, A_i^* is precisely the set of all placements of the reference point O in which B intersects A_i . Thus we can shrink B to the single point O and replace the obstacles with their expanded versions, obtaining a much easier problem to solve. As a matter of fact, the free configuration space is simply the complement of the union of the expanded obstacles (the expanded obstacles are in fact concrete examples of the somewhat vaguer notion of forbidden regions mentioned earlier).

While the conceptual idea is quite simple and can be applied to more general instances (for example, 3D translational motion or translational motion of a nonconvex object), it raises some interesting problems. One is that the expanded obstacles are not necessarily disjoint anymore, so we must analyze the pattern in which they intersect. Kedad, Livne, Pach, and I showed that, fortunately, when B is convex, the boundaries of any pair of expanded obstacles intersect in at most two points (assuming "general position" of the obstacles). Furthermore, it was also shown that for any collection of n closed Jordan curves in the plane, having the property that any pair of them intersect at most twice, the union of their interiors has only a linear number of intersections on its boundary (more precisely, at most $6n - 12$ such points). This implies that the complexity of *FP*, for a translating convex polygon, is only linear, which facilitates the design of a simple divide-and-conquer algorithm for calculating *FP*; the algorithm runs in time $O(n \log^2 n)$.

The single-component approach. If, in the preceding discussion, the translating object B is not convex, it is easy to give examples in which *FP* can have a quadratic number of corners (see Figure 5). However, as pointed out earlier, we do not have to compute the entire *FP*; it suffices to compute just the connected component of *FP* that contains the initial placement of B . The hope is that this component will be

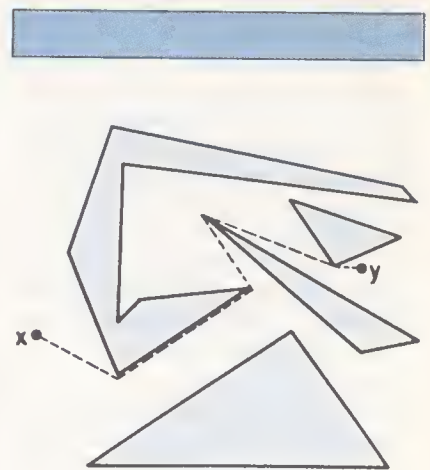


Figure 6. An instance of the 2D Euclidean shortest path problem.

much less complex than the entire *FP* and that consequently it can be calculated by a more efficient algorithm.

In full generality, this amounts to conjecturing that the combinatorial complexity of a single cell in an arrangement of n algebraic surfaces of a small and fixed maximum degree in k -dimensional space is smaller than $O(n^k)$. This conjecture is perhaps one of the most challenging open problems in the theory of motion planning.

For systems with two degrees of freedom, the conjecture was settled in a series of papers showing that the complexity of a single cell in an arrangement of n curves, each pair of which intersect in at most some fixed number s of points, is only slightly superlinear and that it can be calculated in close to linear time. However, very little is known for systems having three or more degrees of freedom. For some systems with three degrees of freedom (for example, a rod in the plane), it is possible to show that the entire *FP* has subcubic complexity. But in almost no instance can one show that the complexity of a single cell is considerably smaller than that of the entire *FP*.

An exception is a result that Aronov and I recently obtained. It shows that when all n constraint surfaces are flat triangles in three-dimensional space, the complexity of a single connected component (cell) in their arrangement is at most about

$O(n^{7/3})$, and that calculation of the cell containing a given point can be accomplished by a randomized algorithm whose expected time complexity is $O(n^{7/3+\delta})$, for any $\delta > 0$. As a corollary, it follows that planning a purely translational motion of a polyhedral object with k faces amidst polyhedral obstacles having a total of n faces (a system obviously having three degrees of freedom) can be accomplished in expected time $O((kn)^{7/3+\delta})$, for any $\delta > 0$.

A mobile convex object moving in a 2D polygonal space. Let B be a convex k -gon moving (translating and rotating) in the plane amidst polygonal obstacles having a total of n corners and edges. The system has three degrees of freedom and is one of the simplest systems arising in practical applications (think of a navigating vehicle, a household or factory mobile server, etc.). Leven and I and Kedem and I¹⁶ studied this particular system recently. We have shown that the entire free configuration space FP has combinatorial complexity $O((kn)^2\beta(kn))$, where $\beta(kn)$ is a very slowly growing function that can be regarded as a constant for any practical value of kn . (Note that the number of collision constraints is $O(kn)$, where each constraint represents placements of B in which either a specific corner of B makes contact with some specific obstacle edge, or a specific side of B touches a specific obstacle corner.) Moreover, using a rather elaborate and sophisticated algorithm, Kedem and I¹⁶ have shown that FP can be calculated in close to quadratic time, specifically in $O((kn)^2\beta(kn) \log kn)$ time.

This algorithm has been implemented recently and tested successfully on commercial equipment. Details of the algorithm are beyond the scope of this article except to note that it uses a mixture of the previous techniques, which involves slicing FP into 2D cross sections FP_θ , in each of which the orientation θ of B is fixed, followed by calculation of these restricted FP s using the techniques reviewed above. We obtain the overall representation of FP by extending the vertices of the 2D cross sections of FP into edges of FP by letting θ vary. Critical values of θ arise when the combinatorial structure of the cross section changes discontinuously. As it turns out, the main (though not exclusive) source of criticalities is placements of B at which it makes three contacts with the obstacles simultaneously (while otherwise remaining free). Leven and I obtained a

previous combinatorial result showing that the maximum number of such free triple contacts is at most $O((kn)^2\beta(kn))$. Using this result, Kedem and I showed that the total complexity of the connectivity "edge graph" representing FP is of the same order of magnitude, and we developed the aforementioned algorithm for calculating that graph.¹⁶

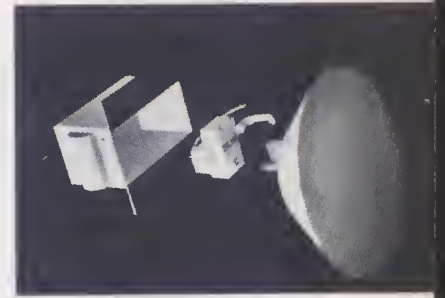
I mention this algorithm also because it is one of the few implemented motion-planning algorithms that conform to the maxim of using exact, nonheuristic, and worst-case efficient techniques. Avnaim, Boissonnat, and Faberjon describe a related recent implementation that solves the motion-planning problem for an arbitrary mobile polygonal robot but has worst-case cubic time complexity (which made it simpler to implement).

Variants of the motion-planning problem

Optimal motion planning. So far the only optimal motion-planning problem studied extensively is that in which the moving system is represented as a single point. The aim in this case is to calculate the Euclidean shortest path connecting the initial and final system placements, given that specified obstacles must be avoided. Most existing work on this problem assumes that the obstacles are either polygonal (in two-dimensional space) or polyhedral (in three-dimensional space).

The 2D case is considerably simpler than the 3D case (see Figure 6). When the free space V in 2D is bounded by n straight edges, we can calculate the desired shortest path in time $O(n^2)$. This is done by constructing a visibility graph VG whose edges connect all pairs of V 's boundary corners that are visible from each other through V , then by searching for a shortest path (in the graph-theoretic sense) through VG . Although the quadratic time bound was improved in certain special cases (the most interesting being when V is the interior of a simple polygon), it is still not known whether shortest paths for a general polygonal space V can be calculated in subquadratic time.

The 3D polyhedral shortest path problem is substantially more difficult. Investigators have shown that finding such a path is NP-hard (an exponential-time procedure for this task is also known). Some special cases in which the path is confined to the surface of a single convex



polyhedron, or must avoid a small number of convex polyhedral obstacles, have also been studied.

A few recent papers studied optimal motion planning for more complex systems, but the results are still fragmentary.

Adaptive and exploratory motion planning. If the environment of obstacles through which motion must be planned is not known to a robot system a priori, but the system is equipped with sensory devices, motion planning assumes a more "exploratory" character. If only tactile (or proximity) sensing is available, then a plausible strategy might be to move along a straight line (in physical or configuration space) directly to the target position, and when an obstacle is reached, to follow its boundary until the original straight line of motion is reached again. Lumelsky and his collaborators² developed and refined this technique, and it has been successful in handling systems with two degrees of freedom—successful, that is, in the sense that this strategy provably reaches the goal, if this is at all possible, with a reasonable bound on the length of the motion. This technique has been implemented on several real and simulated systems and has applications to maze-searching problems.

Attempts to extend this technique to systems with three or more degrees of freedom are currently under way. Cox and Yap recently obtained one such extension, with a somewhat different flavor. They applied the retraction technique to the free configuration space FP and computed within it a certain one-dimensional skeleton N that captures the connectivity of FP . The twist here is that FP is not known in advance, so N must be constructed in an incremental, exploratory manner, adding more features to N as the system explores its environment. Cox and Yap showed

how to implement this exploration in a controlled manner that does not require too many "probing" steps and that enables the system to recognize when the construction of N has been completed (if the goal has not been reached beforehand).

Even when the environment is fully known to the system, other interesting issues arise if the environment is changing. For example, when the robot picks up some of the objects in its environment and moves them, we want fast techniques for incremental updating of the environment model and the data structures used for motion planning. Moreover, whenever the robot grasps an object to move it, robot plus grasped object become a new moving system and might require a different motion-planning algorithm, but one whose relationship to motions of the robot alone needs to be investigated. Adaptive motion-planning problems of this kind remain to be studied.

Motion planning in the presence of moving obstacles. Interesting generalizations of the motion-planning problem arise when some of the obstacles in the robot's environment are assumed to be moving along known trajectories. In this case the robot's goal will be to "dodge" the moving obstacles while moving toward its target placement. In this dynamic motion-planning problem, we can reasonably assume some limit on the robot's velocity and/or acceleration. Two initial studies of this problem by Reif and I and by Sutner and Maass indicate that the problem of avoiding moving obstacles is substantially harder than the corresponding static problem. By using time-related configuration changes to encode Turing machine states, these studies show that the problem is $Pspace$ -hard even for systems with a small and fixed number of degrees of freedom (Canny and Reif have obtained additional negative related results). However, polynomial-time algorithms are available in a few particularly simple special cases.

Miscellaneous other variants. Many other variants of the motion-planning problem have been considered. For lack of space I cannot cover them all in detail, but I would like to mention a few.

One variant deals with *constrained motion planning*, in which the motion of the object B is constrained not to exceed certain velocity or acceleration thresholds.

For example, what is the fastest way for an object to go from an initial placement at some initial velocity to a final placement at some specified final velocity, assuming there is some bound on the maximum acceleration that B can exert? Even without obstacles such problems are far from trivial, and the presence of (stationary or moving) obstacles makes them extremely complicated to solve. Several recent works have addressed these issues, including (1) analysis of the bounded-acceleration case mentioned above, (2) a variant in which the system must avoid stationary obstacles in the plane, and the curvature of the motion path cannot exceed a certain threshold, and (3) an approximate polynomial-time solution for minimum-time motion planning, taking into account acceleration and other constraints.

Another class of problems involves motion planning with uncertainty. Here the moving system has only approximate knowledge of the obstacles' geometry and/or of its own current position and velocity. In these cases the system's main source of "accurate" data is usually sensory feedback upon contact with an obstacle. The objective is to devise a strategy that will guarantee the system's reaching its goal. Such a strategy usually proceeds through a sequence of free motions (until an obstacle is hit) intermixed with compliant motions (sliding along surfaces of contacted obstacles) until it can be ascertained that the goal has been reached.

General task planning. Finally we consider the general problem of task planning. Here the system is given a complex task to perform. For example, it is required to assemble some part from several components or to restructure its work cell into a new layout, but the precise sequence of substeps needed to attain the final goal is not specified and must be determined by the system.

Not much has been done about an exact and efficient algorithmic approach to this general problem, and it is clear that the problem is significantly more complex than the simpler motion-planning problem (some instances of which, as we have seen, are already complex enough). Still, several interesting basic problems can be distilled as special cases of task planning and have recently been investigated. A typical example is the problem of "movable separability of sets," as studied by Toussaint and his collaborators. In this problem we reverse the assembly process, attempting

instead to take apart the final assembled object by moving subcollections of its parts away from one another. We continue this process until all parts have been separated.

Several algorithms have been obtained for various special cases, and many interesting open problems have emerged. For example, consider a collection of convex polyhedra in three-dimensional space (with pairwise disjoint interiors). Can they always be partitioned into two disjoint and nonempty subcollections, so that the polyhedra in one collection can be translated rigidly together in some direction so as to be separated from those in the second collection (which remain stationary and should not collide with the moving polyhedra)? Some counterexamples show that this conjecture is false if we are allowed to move only one polyhedron at a time away from the others.

Another research area related to task planning is that of *grasp planning*, where a multifinger system must plan the grasping of a given object B so as to minimize slippage or to ensure that the fingers avoid interfering with subsequent tasks to be executed by the gripped object. (For example, if B is to be placed on a table, no finger should touch the face of B that is to lie on the table.)

In conclusion, I hope this article has conveyed the mathematical depth and sophistication of the motion-planning problem. While considerable progress has been made in understanding the geometric and combinatorial structure of these problems, many more deep challenges lie ahead. For example, how close are we to obtaining efficient non-heuristic solutions for planning the motion of standard commercial manipulators with six degrees of freedom? I anticipate that in the coming years we will see more of these algorithms implemented and that motion planning will eventually become a standard component of many robot control systems. Judging by the progress made so far, we can hope that real-world implementations of such systems are not too far away. □

Acknowledgments

Work on this article was supported by Office of Naval Research grant N00014-87-K-0129, by National Science Foundation grant NSF-DCR-83-20085, by grants from Digital Equipment Corporation and IBM, and by a grant from the US-Israeli Binational Science Foundation.

References

1. T. Lozano-Perez, "A Simple Motion Planning Algorithm for General Robot Manipulators," tech. report, AI Lab, MIT, Cambridge, Mass., 1986.
2. V. Lumelsky and A. Stepanov, "Path Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmica*, Vol. 2, 1987, pp. 403-430.
3. J.T. Schwartz and M. Sharir, "On the Piano Movers' Problem: I," *Comm. Pure and Applied Math.*, Vol. 36, 1983, pp. 345-398.
4. C. O'Dunlaing, M. Sharir, and C. Yap, "Generalized Voronoi Diagrams for a Ladder: I," *Comm. Pure and Applied Math.*, Vol. 39, 1986, pp. 423-483; and "Generalized Voronoi Diagrams for a Ladder: II," *Algorithmica*, Vol. 2, 1987, pp. 27-59.
5. D. Leven and M. Sharir, "An Efficient and Simple Motion-Planning Algorithm for a Ladder Moving in Two-Dimensional Space Amidst Polygonal Barriers," *J. Algorithms*, Vol. 8, 1987, pp. 192-215.
6. Y. Ke and J. O'Rourke, "Moving a Ladder in Three Dimensions: Upper and Lower Bounds," *Proc. Third ACM Symp. Computational Geometry*, 1987, pp. 136-146.
7. J.T. Schwartz and M. Sharir, "On the Piano Movers' Problem: II," *Advances in Applied Math.*, Vol. 4, 1983, pp. 298-351.
8. J. Canny, *Complexity of Robot Motion Planning*, doctoral dissertation, MIT, Cambridge, Mass., 1987.
9. J. Reif, "Complexity of the Mover's Problem and Generalizations," *Proc. 20th IEEE Symp. Foundations of Computer Science*, CS Press, Los Alamitos, Calif., 1979, pp. 421-427.
10. J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides, "Movement Problems for 2-Dimensional Linkages," *SIAM J. Computing*, Vol. 13, 1984, pp. 610-629.
11. J.E. Hopcroft, J.T. Schwartz, and M. Sharir, "On the Complexity of Motion Planning for Multiple Independent Objects: PSPACE Hardness of the 'Warehouseman's Problem,'" *Int'l J. Robotics Research*, Vol. 3, No. 4, 1984, pp. 76-88.
12. M. Sharir and E. Ariel-Sheffi, "On the Piano Movers' Problem: IV," *Comm. Pure and Applied Math.*, Vol. 37, 1984, pp. 479-493.
13. J.T. Schwartz and M. Sharir, "On the Piano Movers' Problem: V," *Comm. Pure and Applied Math.*, Vol. 37, 1984, pp. 815-848.
14. C. O'Dunlaing and C. Yap, "A 'Retraction' Method for Planning the Motion of a Disc," *J. Algorithms*, Vol. 6, 1985, pp. 104-111.
15. T. Lozano-Perez and M. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Comm. ACM*, Vol. 22, 1979, pp. 560-570.
16. K. Kedem and M. Sharir, "An Efficient Motion Planning Algorithm for a Convex Rigid Polygonal Object in 2-Dimensional Polygonal Space," to be published in *Discrete Computational Geometry*.

(Editor's note: References 3, 4, 7, 9, 10, and 12-14 have been reprinted under one cover: J.E. Hopcroft, J.T. Schwartz, and M. Sharir, eds., *Planning, Geometry, and Complexity of Robot Motion*, Ablex Publishing, Norwood, N.J., 1987.)

Further reading

The literature on motion planning and related problems has grown explosively in the last few years. The additional references below in the various subareas will allow interested readers to obtain a broader view.

Other surveys on motion planning and related topics

Schwartz, J.T., and M. Sharir, "Motion Planning and Related Geometric Algorithms in Robotics," *Proc. Int'l Congress of Mathematicians*, Vol. 2, Berkeley, Calif., Aug. 1986, pp. 1,594-1,611.

Sharir, M., "Efficient Algorithms for Planning Purely Translational Collision-Free Motion in Two and Three Dimensions," *Proc. IEEE Symp. Robotics and Automation*, CS Press, Los Alamitos, Calif., 1987, pp. 1,326-1,331.



FULL AT&T C++: ANNOUNCING VERSION 1.2

Guidelines announces its port of **version 1.2** of AT&T's C++ translator. As an object-oriented language C++ includes: classes, inheritance, member functions, constructors and destructors, data hiding, and data abstraction. "Object-oriented" means that C++ code is more readable, more reliable and more reusable. And that means faster development, easier maintenance, and the ability to handle more complex projects. C++ is **Bell Labs' answer to Ada and Modula 2**. C++ will more than pay for itself in saved development time on your next project.

C++

from **GUIDELINES** for the IBM PC: \$295

Requires IBM PC/XT/AT or compatible with 640K and a hard disk.

Note: C++ is a translator, and requires the use of Microsoft C 3.0 or later.

Here is what you get for \$295:

- The full AT&T v1.2 C++ translator.
- Libraries for stream I/O and complex math.
- **The C++ Programming Language**, the definitive 327-page tutorial and description by Bjarne Stroustrup, designer of C++.
- Sample programs written in C++.
- Improved installation guide and documentation.
- 30-day money-back guarantee.

NOW AVAILABLE FOR UNIX V/386 - \$495

To Order:

send check or money order to:

GUIDELINES SOFTWARE, INC.
P. O. Box 749, Dept. CT
Orinda, CA 94563

To order with VISA or MC,
phone (415) 254-9183.
(CA residents add 6% tax.)

C++ is ported to the PC by Guidelines under license from AT&T.
Call or write for a free C++ information package.

Sharir, M., "Davenport Schinzel Sequences and Their Geometric Applications," in *Theoretical Foundations of Computer Graphics and CAD*, R.A. Earnshaw, ed., NATO ASI Series, Vol. F40, Springer-Verlag, Berlin, 1988, pp. 253-278.

Yap, C.K., "Algorithmic Motion Planning," in *Advances in Robotics*, Vol. 1, J.T. Schwartz and C.K. Yap, eds., Lawrence Erlbaum, Hillsdale, N.J., 1987, pp. 95-143.

Implementations and heuristic approaches to motion planning

Avnaim, F., J.D. Boissonnat, and B. Faberjon, "A Practical Exact Motion Planning Algorithm for Polygonal Objects Amidst Polygonal Obstacles," *Proc. IEEE Symp. Robotics and Automation*, CS Press, Los Alamitos, Calif., 1988, pp. 1,656-1,661.

Brooks, R.A., "Solving the Find-Path Problem by Good Representation of Free Space," *IEEE Trans. Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, 1983, pp. 190-197.

Brooks, R.A., "Planning Collision-Free Motions for Pick-and-Place Operations," *Int'l J. Robotics Research*, Vol. 2, No. 4, 1983, pp. 19-40.

Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Int'l J. Robotics Research*, Vol. 5, No. 1, 1986, pp. 90-98.

Lozano-Pérez, T., "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Computers*, Vol. C-32, No. 2, 1983, pp. 108-119.

Moravec, H.P., *Robot Rover Visual Navigation*, doctoral dissertation, Stanford Univ., UMI Research Press, 1981.

Udupa, S., *Collision Detection and Avoidance in Computer-Controlled Manipulators*, doctoral dissertation, California Inst. of Technology, Pasadena, Calif., 1977.

More references to standard motion-planning problems

Agarwal, P., and M. Sharir, "Red-Blue Intersection Detection Algorithms with Applications to Motion Planning and Collision Detection," *Proc. Fourth ACM Symp. Computational Geometry*, 1988, pp. 70-80.

Aronov, B., and M. Sharir, "Triangles in Space, or: Building (and Analyzing) Castles in the Air," *Proc. Fourth ACM Symp. Computational Geometry*, 1988, pp. 381-391.

Guibas, L., M. Sharir, and S. Sifrony, "On the General Motion Planning Problem with Two Degrees of Freedom," *Proc. Fourth ACM Symp. Computational Geometry*, 1988, pp. 289-298.

Kedem, K., et al., "On the Union of Jordan Regions and Collision-Free Translational Motion Amidst Polygonal Obstacles," *Discrete and Computational Geometry*, Vol. 1, 1986, pp. 59-71.

Sharir, M., and S. Sifrony, "Coordinated Motion Planning for Two Independent Robots," *Proc. Fourth ACM Symp. Computational Geometry*, 1988, pp. 319-328.

Sifrony, S., and M. Sharir, "An Efficient Motion Planning Algorithm for a Rod Moving in Two-Dimensional Polygonal Space," *Algorithmica*, Vol. 2, 1987, pp. 367-402.

Variants of the motion-planning problem

Canny, J., B. Donald, and J. Reif, "On the Complexity of Kinodynamic Planning," *Proc. 29th IEEE Symp. Foundations of Computer Science*, CS Press, Los Alamitos, Calif., 1988, pp. 306-318.

Canny, J., and J. Reif, "New Lower Bound Techniques for Robot Motion Planning Problems," *Proc. 28th IEEE Symp. Foundations of Computer Science*, CS Press, Los Alamitos, Calif., 1987, pp. 49-60.

Cox, J., and C.K. Yap, "On-Line Motion Planning: Moving a Planar Arm by Probing an Unknown Environment," tech. report, Courant Inst., New York Univ., N.Y., July 1988.

Fortune, S., and G. Wilfong, "Planning Constrained Motion," *Proc. 20th ACM Symp. Theory of Computing*, 1988, pp. 445-459.

Mishra, B., J.T. Schwartz, and M. Sharir, "On the Existence and Synthesis of Multifinger Positive Grips," *Algorithmica*, Vol. 2, 1987, pp. 541-558.

Natarajan, B.K., "On Planning Assemblies," *Proc. Fourth ACM Symp. Computational Geometry*, 1988, pp. 299-308.

Papadimitriou, C., and E. Silverberg, "Optimal Piecewise Linear Motion of an Object among Obstacles," *Algorithmica*, Vol. 2, 1987, pp. 523-539.

Reif, J., and M. Sharir, "Motion Planning in the Presence of Moving Obstacles," *Proc. 26th IEEE Symp. Foundations of Computer Science*, CS Press, Los Alamitos, Calif., 1985, pp. 144-154.

Manager of Technical Training

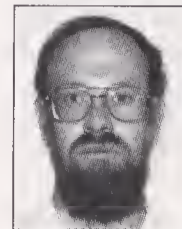
Quotron — one of the world's largest, most reliable suppliers of on-line financial information services — is seeking an individual to be responsible for developing, sourcing and delivering cost-effective technical training programs, in support of our Systems Development Operation. Position involves assessing technical skills requirements and building/managing a technical training staff.

Requirements include 10+ years experience in the technical education field with proven, hands-on technical training expertise. Experience in UNIX*, real-time systems and networks a plus. Strong leadership and interpersonal skills are essential, as well as budget management ability.

Quotron provides an aggressive compensation and benefits package, which includes up to 3 weeks vacation during the first year of employment and a corporate fitness center. For consideration, send resume to P. Brown at: Quotron Systems, Inc., P.O. Box 92759, World Way Postal Center, Los Angeles, CA 90009. Equal Opportunity Employer.

*UNIX is a trademark of AT&T Bell Labs.

Quotron



Micha Sharir is a visiting professor of computer science and associate director of the Robotics Research Laboratory at the Courant Institute of Mathematical Sciences, New York University. He is also a professor of computer science at Tel Aviv University in Israel. His research interests include motion planning in robotics, computer vision, and computational geometry. He received BS, MS, and PhD degrees in mathematics from Tel Aviv University in 1970, 1972, and 1976, respectively.

Readers can contact the author at New York University, Robotics Laboratory, 719 Broadway, New York, NY 10003.

Task-Level Planning of Pick-and-Place Robot Motions

Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell
MIT Artificial Intelligence Laboratory



A task-level robot system is one that can be instructed in terms of task-level goals, such as "Grasp part A and place it inside box B." This type of specification contrasts sharply with that required for existing industrial robot systems, which insist on a complete specification of each motion of the robot and not simply a description of a desired goal. An important characteristic of task-level specifications is that they are independent of the robot performing the task, whereas a motion specification is wedded to a specific robot.

Task-level robot systems have long been a goal of robotics research. As early as 1961, Ernst's PhD thesis at MIT¹ attempted to develop such a system. Since then, a variety of task-level robot systems have been proposed, and several of them have seen some level of implementation. Lozano-Pérez and Taylor provide a survey of previous work in this area.²

For the past three years, we have been developing a task-level robot system named Handey. The current system is by no means a complete task-level system; it is limited to *pick-and-place* operations, that is, picking up a part and placing it at a specified destination. The current implementation has successfully carried out dozens of pick-and-place operations involving a variety of parts in relatively complex



**The Handey system
breaks the
pick-and-place problem
into nearly independent,
computationally
feasible subproblems as
a step toward a
comprehensive
task-level system.**

environments. Figure 1 shows a sequence of intermediate steps of a pick-and-place operation planned by Handey from a specification of the desired final position of the part and geometric and kinematic models of the robot and the environment. The steps illustrated in the figure are

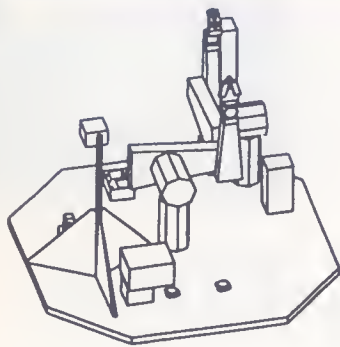
- (1) the initial position of the robot and the parts,
- (2) grasping a part at a location chosen by the system to avoid collisions

- with nearby objects,
- (3) placing the part on the table,
- (4) regrasping the part at a new location compatible with the environment at the destination,
- (5) placing the part at the specified destination, and returning to the initial position.

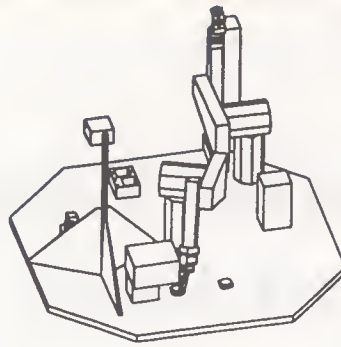
The pick-and-place problem

Consider the overall problem faced by a task-level robot system, given a mechanical assembly task and a single, sufficiently precise robot. (In this article, we assume that the position of all the parts are known to high accuracy and the robot's positioning accuracy is sufficient to carry out the assembly simply by moving the part to a fixed target position. If this assumption is not satisfied, then assembly planning must include *fine-motion* planning, that is, the synthesis of motion strategies that can achieve a desired goal in the presence of sensing and control error.^{3,4}) We can summarize the overall assembly planning problem, of which pick-and-place is but a part, as follows:

- Choose an order of assembly for the parts.
- Choose initial positions for all the parts.



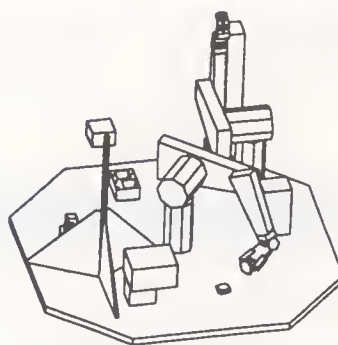
Step 1



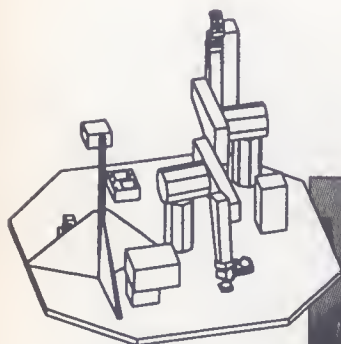
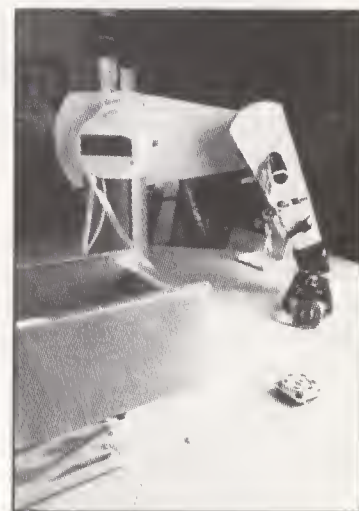
Step 2



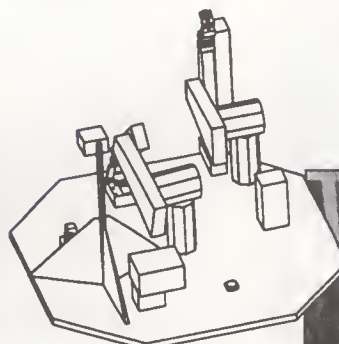
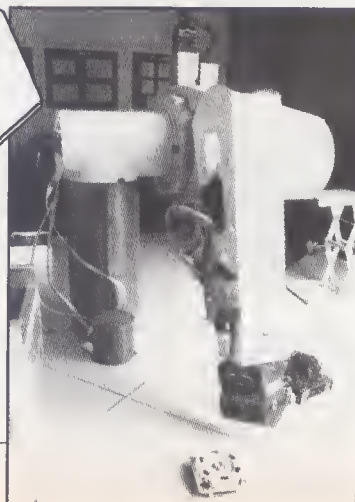
Figure 1. Steps 1-5 in a pick-and-place operation planned by Handey.
Photographs courtesy of Francisco Garcini.



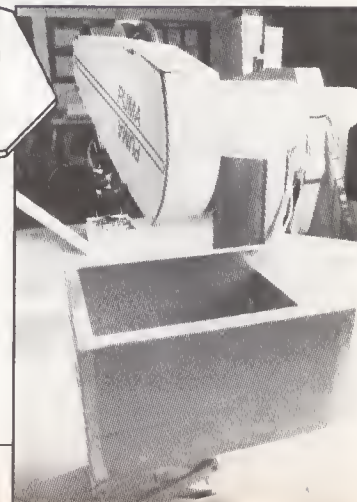
Step 3



Step 4



Step 5



Then, for each assembly step:

- Choose a grasp on the part.
- Plan a motion to grasp the part.
- Plan a motion to the assembly location for the part.
- Plan a motion to extract the gripper.

The pick-and-place problem corresponds to the last four steps above.

Ideally, the assembly problem will reduce to a number of independent subproblems, roughly one per line in the characterization above. Unfortunately, this is only true in extremely simple cases. In fact, the solutions to all these problems are tightly linked:

(1) The optimal initial placement of the parts depends on the order of assembly.

(2) The order of assembly depends on the feasibility of certain assembly steps, for example, whether a subassembly can be inserted into the main assembly in one piece or whether it needs to be assembled in position.

(3) The choice of a grasp on a part depends on the environment at the initial

location, which depends on the initial placement of nearby parts. The grasp also depends on the placement of parts at the target, which depends on what previous assembly steps have been performed. The choice of grasp depends on the range of motions of the robot; surprisingly many potential grasps simply will not be reachable.

(4) The choice of a grasp (and ungrasp) motion depends on the choice of a grasp and on the whole environment, especially near the grasp and destination.

(5) The choice of an assembly motion depends on whether a grasp compatible with the assembly has been found; otherwise, the part must be regrasped.

In short, all the subproblems of the assembly problem are interdependent. This also holds true for the pick-and-place problem we are considering.

One possible approach to the pick-and-place problem, given the interdependence of the decisions, is to treat it as a single motion-planning problem with special

constraints. In particular, the effective shape of the robot changes once it grasps the object, so the constraints on its motion change. The problem with this approach is that it runs into significant computational complexity; the additional degrees of freedom in the choice of a grasp add to the robot's motion freedoms. But, we expect all complete algorithms for motion planning to have worst-case time-complexity that is exponential in the degrees of free-



Configuration space

Robot manipulators are articulated devices made up of a series of rigid links connected by one-degree-of-freedom joints. Joint motions are either rotational or translational. The positions of all parts of a rigid robot are completely specified by the values of the joint parameters, known collectively as the *joint angles*. Many robots support an alternative specification for desired position, namely, the position and orientation of the robot gripper. But, gripper position is not a unique specification of the complete robot position; many sets of joint angles can exist that place the gripper in the same location.

Any set of parameters that uniquely specify the position of every part of a system is called a *configuration*, and the space defined by those parameters is the *configuration space* or *C-space*. Most algorithmic approaches to motion planning require a characterization of those configurations of the robot that cause collisions (or the complement of that set). We call the collection of configurations that produce collisions the *C-space obstacles*.⁸

Consider the following simple case (see Figure 2a). In this example, we

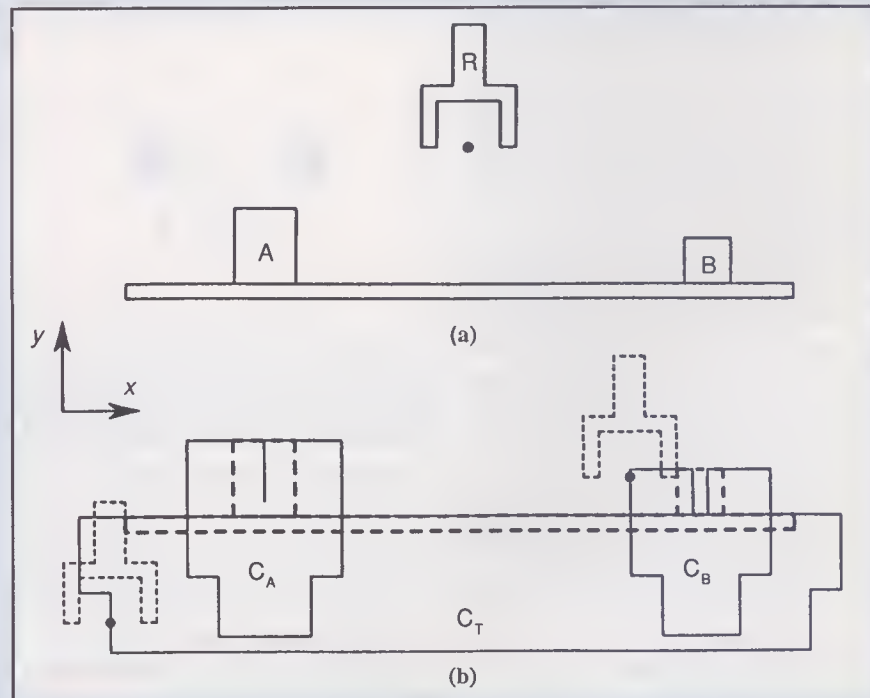


Figure 2. Cartesian and C-space obstacles. (a) A scene with three obstacles, A, B, and the table. The disembodied robot hand R is capable only of displacements along the x and y axes. (b) The C-space for this hand is the x,y plane. The C-space obstacles corresponding to A (C_A), B (C_B), and the table (C_T) are seen. The gripper is shown (in a fine dashed line) at two points of the C-space obstacle boundary.

dom but polynomial in the size of the environment (Reif has shown motion planning to be PSPACE-hard⁵). Therefore, any practical solution to the pick-and-place and assembly problems must involve decoupling and other forms of dimensionality reduction (see the accompanying sidebar, "Configuration space").

Approximate approaches to the pick-and-place problem

The crucial step in solving the pick-and-place problem is choosing how to grasp the part. Once this choice has been made, the problem boils down to separate motion-planning problems in the robot's C-space. But, we cannot choose a grasp simply by looking at the environment near the part; we must consider the effect of the grasp on the possibility of finding a path to the goal. Therefore, any attempt at decoupling must consider constraints imposed on the choice

of grasp by the environment at the goal as well as that at the origin.

One necessary condition that the choice of grasp point must satisfy is that there be no collisions of the robot with any object at either the initial or final position of the part. In addition, there must be some path connecting these initial and final positions along which there are no collisions. In practical situations, however, given safe initial and final destinations, a path can usually be found.

Given this heuristic assumption about the availability of paths, there only remains the problem of finding a grasp that is reachable for the part in its initial position and avoids collisions for the initial and final positions. There are at least two ways of doing this:

(1) Characterize the reachable grasps at the initial grasp position, then characterize the grasps that cause no collisions at the destination. The grasps in the intersection of these two grasp sets are collision-free at both positions and reachable at the initial position.

(2) Compute the transformation T that maps the grasped part from its initial to its final location. Apply the inverse transformation T^{-1} to a copy of the obstacles near the final position of the part. Add these transformed obstacles to the obstacles near the initial position of the part. Find a path to any legal grasp that avoids both sets of obstacles (see Figure 4).

These two methods are not equivalent; the second method will, in general, find fewer grasps than the first method will. In finding a path that avoids both sets of obstacles, we have constrained the problem beyond what is strictly necessary to guarantee finding a grasp that is reachable at the initial position and safe at the final one. Recall that we are planning the initial grasp approach and not the path to the final destination. It is not necessary, therefore, that the complete approach path avoid the obstacles derived from the destination, only that the *chosen grasp*, that is, the final position in the approach path, avoid the obstacles.

We chose the second method for Handey

are limited to a disembodied robot hand capable only of displacements along the x and y axes; we assume that the finger opening remains fixed to simplify the C-space. The C-space for this hand is the x, y plane. The C-space obstacles corresponding to objects A and B and the table in Figure 2a are the objects bounded by dark lines in Figure 2b. Every x, y point inside one of the C-space obstacles represents an x, y position of R 's reference point (the dark circle in Figure 2a) that causes a collision. C_A are the configurations that cause a collision between R and A ; C_B are the configurations that cause a collision between R and B ; C_T are the configurations that cause a collision between R and the table.

Consider a simple robot with two rotational joints, whose joint angles are θ_1 and θ_2 (see Figure 3a). A point in the C-space specifies both joint angles and, therefore, the position of every part of the robot. The set of configurations for which the robot is in collision with an obstacle defines C-space obstacles. In this sense, the C-space for the robot is analogous to that of the disembodied hand above, but the C-space obstacles of the robot are significantly more complex than the ones for the hand,⁸ (see Figure 3b). Note that the C-space parameters of the ro-

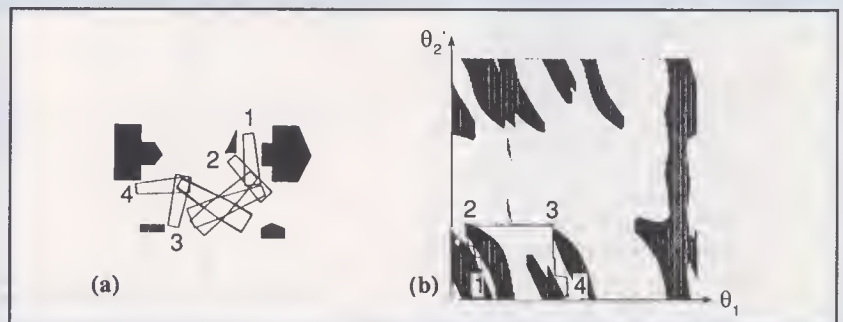


Figure 3. C-space obstacles for a simple robot. (a) A robot with two joint angles, θ_1 and θ_2 , and its obstacles. (b) The corresponding C-space obstacles (the hatched regions). The joint angles are also the labels on the axes of the C-space. A safe path in this C-space is shown, together with four configurations of the arm along the path.

bot are angles. Therefore, the line at the top of the diagram, $\theta_2 = 2\pi$, is the same line as the bottom of the diagram, $\theta_2 = 0$; we can say the same for the left and right lines, $\theta_1 = 2\pi = 0$. We can wrap the diagram into a tube so that the top and bottom lines meet, then we can wrap that tube into a doughnut (torus) to make the left and right ends meet. Thus, the C-space for the two-link robot is really the surface of a torus.

Finding a collision-free path between two specified configurations requires finding some path in the C-space that connects the two specified configurations and does not penetrate any of the C-space obstacles. Doing this requires characterizing the space outside of all of the C-space obstacles, possibly by a decomposition into disjoint cells. A sample path between the two arm configurations shown in Figure 3a is shown in the C-space of Figure 3b.

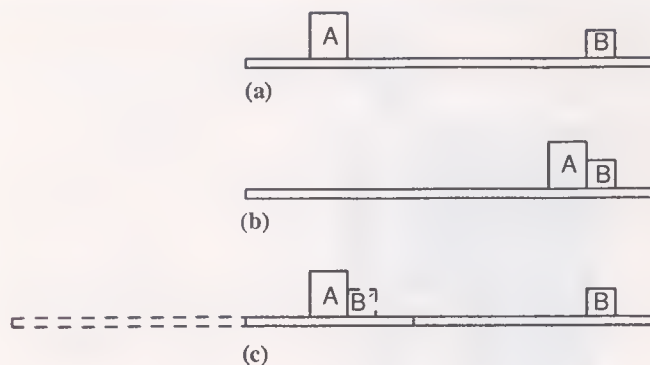


Figure 4. A simple pick-and-place problem: (a) the initial position of A; (b) the final position of A; and (c) the pick-and-place example showing the obstacles for the final position of part A (dashed lines) superimposed on the obstacles for the initial position (solid lines).

because it does not require characterizing sets of grasps, whether simply collision-free or reachable. Computing these sets can be a burdensome task. The second method requires only the ability to find a path from a known initial position to some point within a constrained set (legal grasps).

In many cases, no grasp exists with an approach path that avoids obstacles from both the origin and the destination. In this case, Handey plans a sequence of regrasping steps (placing the object on the table and then regrasping it) that results in a grasp compatible with the target position.⁶ Once again, we use only the necessary condition that the grasp does not cause a collision at the origin or destination.

Here is a more complete outline of the solution of the pick-and-place problem in Handey:

(1) Enumerate the possible distinct types of grasps. Handey operates in the domain of polyhedral models and robots with parallel jaw grippers; therefore, the distinct grasps are limited to pairs of part features (either faces, edges, or vertices) that can be in simultaneous contact with the parallel interior faces of the gripper. Our implementation is currently limited to pairs of parallel faces.

(2) Rank the potential grasp faces by the percentage of the area of intersection of the faces that remains unobstructed after projecting onto the face any obstacles near

the face, at both origin and destination.

(3) Select a potential pair of grasp faces, and pick target points for the fingertips in the unobstructed sections of the faces.

(4) Plan a path for the robot gripper, constrained to move parallel to the grasp faces, from a nearby safe point (chosen arbitrarily) to a point close to the chosen grasp point for which there is sufficient overlap between the gripper and the grasp faces. If no path can be found, select a different grasp face and try again. The path must avoid collisions and keep the robot within its legal range of motion.

(5) Plan a collision-free motion (using a low-resolution approximation of C-space) for the robot from its current position to the arbitrarily chosen point near the initial grasp.

(6) Plan a collision-free motion for the robot, carrying the grasped part (using a low-resolution approximation of C-space) from the initial grasp position to the specified target position.

(7) When it is not possible to find a grasp that avoids collisions at the initial and target positions, plan a sequence of regrasping steps that produce a grasp compatible with the target position. For each required grasp and placement, plan collision-free motions for the robot.

The crucial steps in this formulation rely on finding collision-free paths between two known points or between a known point and a target set.



The worst-case complexity of motion planning is exponential in the number of degrees of freedom of the robot; the best motion-planning algorithm for a d -degree-of-freedom robot has time complexity $O(n^d \log n)$, where n is the product of the number of faces, edges, and vertices in the obstacles.⁷ Although this bound is polynomial in the environment size, n , this complexity still makes complete motion-planning algorithms for real robots impractical on typical serial computers. We also found that pick-and-place planning has a higher dimensionality than simple motion planning. Therefore, we have pursued a heuristic decoupling strategy for pick-and-place planning. To obtain good running times, we must also pursue heuristic strategies into the motion-planning subproblems.

Heuristic motion planning in Handey

Earlier, we outlined an approach to motion planning based on computing C-space obstacles; Lozano-Pérez describes this approach in more detail.⁸ Although we could use the approximate algorithms described there to solve directly the six-degree-of-freedom motion-planning problems required for the typical pick-and-place problem, the running times would be too large for practical use, even in experimentation. Instead, we have adopted several heuristic methods of reducing both the dimensionality and the average size of these motion-planning problems:

(1) Using a local motion planner for small motions near obstacles together with a low-resolution C-space for large motions farther away from obstacles. The local planner simulates the effect of a body being acted on by repulsive forces from the obstacles and attractive forces from the goal. This general approach is known as the *artificial-potential-field approach*.⁹

(2) Limiting the large motions to the

first three joints of the robot, but with the ability to change the values of the last three joints.

(3) Approximating arbitrary polyhedral obstacles by obstacles with a constant cross section that simplify the computation of the low-resolution C-space obstacles.

The combination of these strategies has significantly reduced the running times for motion planning (from a few minutes to perhaps 30 seconds) without significantly reducing Handey's ability to solve pick-and-place problems. We describe these methods in the following subsections.

Local planning: the quasipotential method. In Handey, motion near an obstacle happens primarily when the object is being grasped. In those circumstances, the robot's motion is constrained so that the gripper moves in the plane of the chosen grasp faces. This means that we have a three-degree-of-freedom planning problem instead of a six-degree-of-freedom problem. But, those three motion freedoms correspond to Cartesian motions of the gripper in the grasp plane (x, y, θ) and not to individual joint motions of the robot arm. Exploiting this reduction in dimensionality requires a different planner from the one we use for the gross motions of the arm.

We could construct approximate x, y, θ C-space obstacles for the gripper moving in the grasp plane and plan the gripper motions in this space. But, instead, we have adopted a local, incremental planner that does not build an explicit C-space. We chose this strategy for two reasons:

(1) In Handey, some of the obstacles might not be modeled directly as polyhedra by the system; instead, they might be represented implicitly by an array of depth measurements. We could have tried to build a polyhedral approximation to this data and use the existing C-space obstacle algorithms for polyhedra. Instead, we chose a method that deals with the data more directly.

(2) Typical cases of approaching obstacles within the grasp plane are simple, and a nearly direct path exists. Incremental methods have a lower computational overhead in these situations. In difficult cases, local, incremental methods fail to find an answer. In those cases, we can fall back on guaranteed methods.

Traditional potential methods⁹ measure the distance between a number of points on the moving body (the gripper in our case) and obstructions in the grasp plane, com-

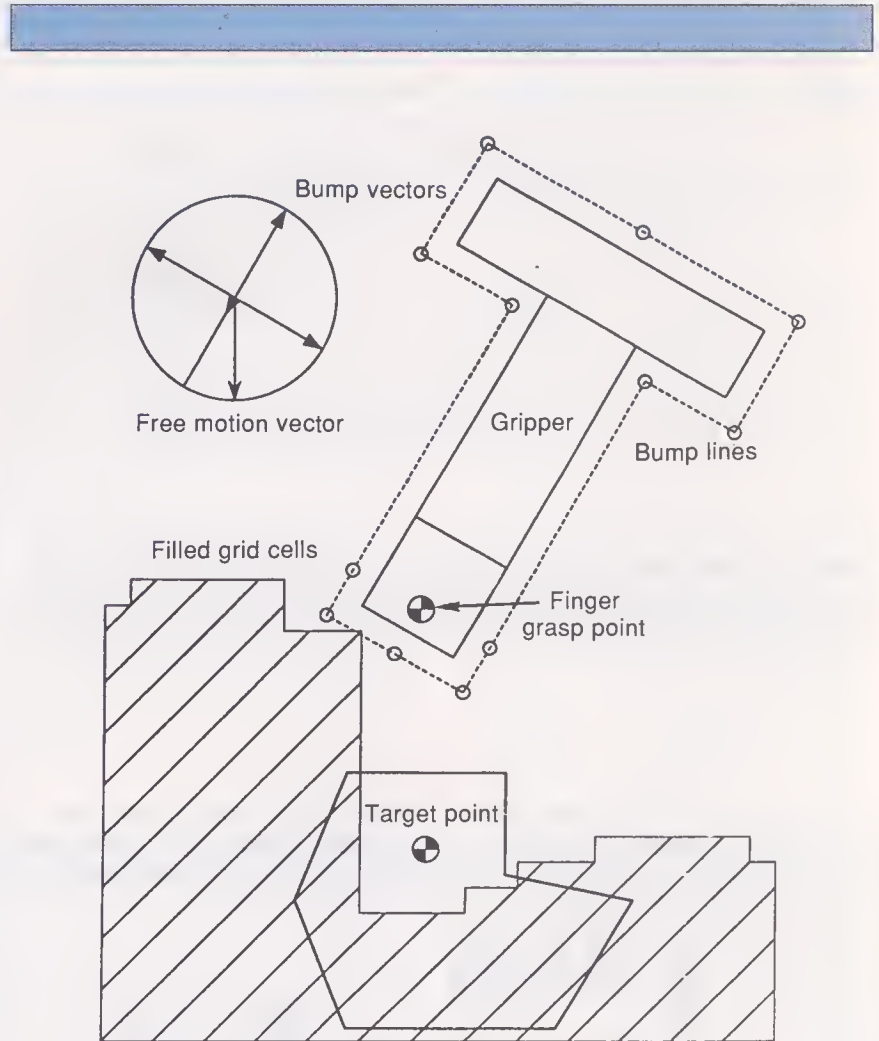


Figure 5. Illustration of the definitions in the quasipotential method.

pute repulsive forces proportional to a power of the distance, and sum them with an attractive force based on the distance from the gripper to the goal. These total artificial forces and torques acting on the gripper are then used to compute a motion for the gripper via the simulated dynamics of a viscous damper, $f = bv$, where f is the total force/torque vector and v is the resulting velocity vector.

Our version of this method effectively uses a potential that is a high power of the distance. Beyond some gripper-to-obstacle distance d , the force on the manipulator is 0; within that distance, the force is such as to prohibit motion toward the obstacle. A grid represents the motion plane, and obstacles correspond to filled cells in the grid. The starting point is some point on

the edge of the grid in the same connected component of the grid as the target point.

Surrounding the gripper at a distance d and moving with it are bump lines, that is, line segments on the grasp plane that are checked each iteration for collisions with filled grid cells (see Figure 5). A bump vector is a unit vector perpendicular to a bump line pointing away from the gripper.

In the absence of intervening filled grid cells in the motion plane, the gripper's motion will be a simple translation along the vector connecting the finger and target point. We call the unit vector in this direction the goal vector.

After investigating all the bump lines for collisions with filled cells, we construct a unit circle and map onto it the goal vector and the bump vectors whose correspond-

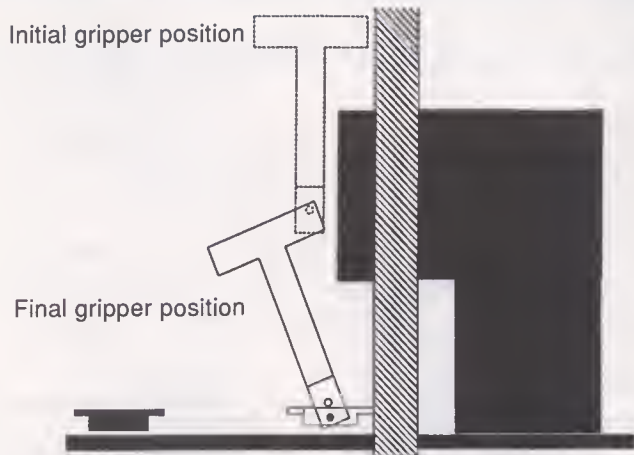


Figure 6. The initial and final position of the gripper for the initial grasp of the example in Figure 1. The dark obstacles are present at the pickup point, the hatched obstacle is a copy of the table at the destination where the regrasp is to be done.

ing bump lines have not detected collisions. Figure 5 shows a typical situation.

To compute the translational motion of the gripper, we compare the goal vector with the multiplied bump vectors. If the goal vector has no component in the direction of a zero-length bump vector, the gripper moves in that direction. Otherwise, as in the figure, the gripper moves along the nonzero bump vector closest in direction to the goal vector.

The bump lines also provide a convenient way to compute a torque to rotate the gripper. Any colliding bump line produces a torque whose magnitude is proportional to the cross product of the bump vector and a vector connecting the *finger grasp point* with the center of the bump line. The total torque on the gripper is just the sum of the torques generated by each colliding bump line.

The gripper is free to rotate about the finger grasp point (shown near the finger tip in Figure 5). Rotations and translations are limited in such a way that no point on the gripper moves by more than one grid cell per iteration. This precludes the possibility that any filled grid cells will penetrate the bump lines during an incremental motion.

At each iteration, a check ensures that a motion in the computed direction will yield

a reachable gripper position. If the position is not feasible, then a different direction (if any remain) is tried.

Every few iterations, the position of the gripper is compared to a previous position. If no significant progress has been made, the path is terminated. At this point, the position of the gripper in relation to the target area is checked. If the fingers sufficiently overlap the target (for example, the chosen faces of a grasp), then the path is accepted and returned; otherwise, the planner either tries to plan a path from another starting point or tries another pair of faces.

Figure 6 shows initial and final positions obtained by the quasipotential method for the initial grasp of the example in Figure 1. Note that the finger grasp point cannot actually reach the chosen target point, but that the grasp sufficiently overlaps the target face.

Although this method does not guarantee a solution, it has performed quite well in most of the cases we have encountered. We believe we can enhance the method's performance by improving the choice of starting point and avoiding a reliance on a fixed attraction point. We are currently experimenting with a variation of the method that picks a starting point by finding the maximal clearance translational paths from the target point to the edges of



the motion grid. The starting points are the ends of the paths. During each iteration, the attraction point used for computing the artificial forces acting on the gripper will move along the chosen path. We hope this will avoid many of the problems of local minima and nonconvex obstacles.

Global planning: low-resolution C-space. The local quasipotential method is effective near the target when we know that a nearly direct path exists. Local methods are less effective for gross motions that span a significant segment of the work space, so we use a low-resolution C-space method for these motions. Although the basic method is that described in Lozano-Pérez,⁸ the implemented method uses several heuristics to lower the average computation time.

One key approximation limits the C-space obstacle construction to the first three joint angles of the robot. The planner builds three three-dimensional *slices* of the underlying six-dimensional C-space. One slice is built with the wrist angles fixed at their value at the start of the path, another slice with the angles set at their value at the end of the path, and the last slice for the range of wrist angles between the start and the end. The free-space representation in these three slices is linked into a single free-space representation that can be searched for a path.

We build the first two slices at high resolution (quantization of one degree), but make no attempt to build the full three-dimensional slice of the C-space. In fact, we make every attempt to limit the size of the slices. The idea is to move as close as possible to the initial and goal points within the slice that spans the orientations of the wrist between the initial and goal points. Then, only the final segments need to be traveled in the remaining two slices. The computation of these slices, therefore, proceeds on an "on-demand" basis. First, we build the slice for a very narrow range

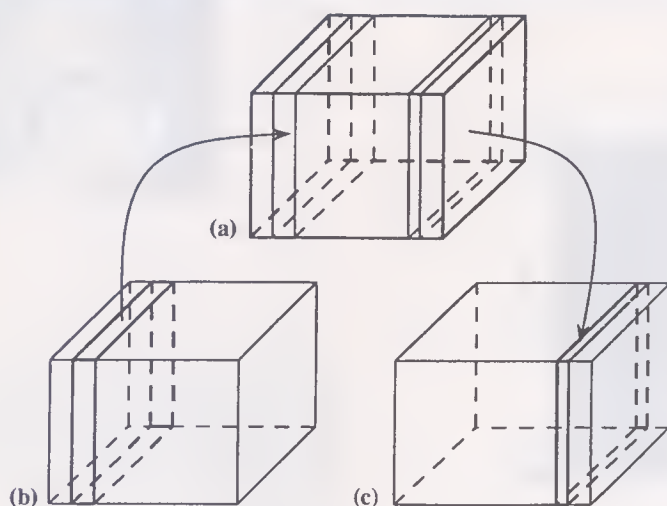


Figure 7. The gross-motion-planning problem is split into three three-degree-of-freedom slices: (a) a low-resolution C-space slice for the total range of values of the wrist angles, (b) a high-resolution C-space slice for the initial values of the wrist angles, and (c) a high-resolution C-space slice for the final values of the wrist angles.

of the first three angles. We stop if we find a path in this range; otherwise, we widen the range incrementally until we find a path. This strategy is illustrated in Figure 7.

The third slice, spanning the range of wrist angles for the motion, is built to span the whole range of the first three joint angles. But, we build this slice at lower resolution (eight degrees) and approximate the robot and the obstacles to simplify the computation. This simplification exploits the fact that the computation of C-space obstacles for planar robots and obstacles is much more efficient than that for solid polyhedra.⁸ We can take advantage of the fact that the Unimation Puma 560 robot used in Handey has, as do many commercial robots, two links with parallel rotation axes (see Figure 8). Therefore, we can approximate the arm as a planar two-link arm operating in a plane determined by the first joint angle. Then, we can approximate the obstacles by tori, centered at the base of the robot, with constant polygonal cross sections in the plane of the second and third links of the robot (see Figure 8).

The result of this process approximates

the three-degree-of-freedom C-space using only planar computations. Although this does not affect the asymptotic complexity of the method, it results in a significant speed up.

The interaction between local and global planning. Given a problem that requires moving from a position close to an obstacle to another position close to an obstacle, such as a regrasping step, Handey proceeds as follows:

- (1) Use the quasipotential planner to move away from the initial position very near an obstacle. Call this point I' .
- (2) Use the quasipotential planner to move away from the final position very near an obstacle. Call this point F' .
- (3) Use the C-space planner to move from I' to F' .

The current implementation of Handey does not choose the points I' and F' with the gross motion planner in mind. In particular, there is no guarantee that these points will be in the low-resolution free-C-space built by the gross motion planner. This is not a requirement, but it minimizes the computation required to plan the com-

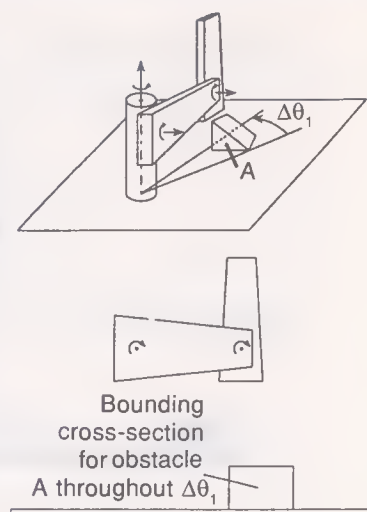


Figure 8. Obstacles can be approximated by tori, centered at the base of the robot, with constant polygonal cross sections in the plane of the second and third links of the robot.

plete path. We could make this connection by marking as desirable targets those grid points (specifying the gripper position and orientation) that map into free-C-space points (specifying the first three joint angles of the arm). Nevertheless, this hybrid local/global approach to planning motions has proven quite effective in our experiments.

Handey is unique for being the first task-level system that has been extensively tested in relatively complex tasks. Since Handey operates by locating an obstructed object placed in a random orientation and taking it to any specified location, it must handle quite general motion-planning problems. But, to be thoroughly tested, it must run in reasonable time. This conflict has forced us to search for simplifying assumptions that do not significantly reduce the system's generality. We believe that this search will be critical to the development of task-level robot systems. Handey's capabilities are only a small fraction of those required for a comprehensive system. Other important capabilities are

- planning force-controlled motions to assemble objects in the presence of position error;
- planning coordinated motions for multiple robots;
- planning the placement of the parts in the work space to optimize task execution; and
- planning the nature and order of operations required to carry out a task, for example, the order in which to assemble parts.

We are currently investigating all these capabilities within the framework of Handey. □

Acknowledgments

This work was funded primarily by the Office of Naval Research under contracts N00014-85-K-0214 and N00014-86-K-0685. Additional support was provided by a National Science Foundation Presidential Young Investigator Award (Lozano-Pérez), the French Centre National de la Recherche Scientifique (Mazer), and Digital Equipment Corporation. In addition to the authors, the following people have contributed to the development of Handey: Pierre Tournassoud of INRIA (France), Alain Lanusse of ETCA (France), and Eric Grimson of MIT. The visits of Lanusse and Tournassoud to MIT were funded by ETCA and INRIA, respectively.

References

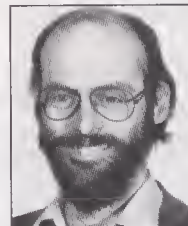
1. H.A. Ernst, *A Computer-Controlled Mechanical Hand*, PhD thesis, MIT, Cambridge, Mass., 1961.
2. T. Lozano-Pérez and R.H. Taylor, "Geometric Issues in Planning Robot Tasks," to be published in *Problems of Robotics*, MIT Press, Cambridge, Mass., 1989.
3. M.T. Mason, "Automatic Planning of Fine Motions: Correctness and Completeness," *Proc. IEEE Int'l Conf. Robotics and Automation*, 1984, CS Press, Los Alamitos, Calif., Order No. 526, pp. 492-503.
4. D.E. Whitney, "Quasistatic Assembly of Compliantly Supported Rigid Parts," *ASME J. Dynamic Systems, Measurement, and Control*, Vol. 104, March 1982, pp. 65-77.
5. J.H. Reif, "Complexity of the Generalized Mover's Problem," in *Planning, Geometry, and Complexity of Robot Motion*, Ablex Publishing, 1987, pp. 267-281.
6. P. Tournassoud, T. Lozano-Pérez, and E. Mazer, "Regrasping," *Proc. IEEE Int'l Conf. Robotics and Automation*, 1987, CS Press, Los Alamitos, Calif., Order No. 787, pp. 1,924-1,928.

7. J.F. Canny, *The Complexity of Robot-Motion Planning*, MIT Press, Cambridge, Mass., 1987.
8. T. Lozano-Pérez, "A Simple Motion-Planning Algorithm for General Robot Manipulators," *IEEE J. Robotics and Automation*, Vol. 3, No. 3, June 1987, pp. 224-238.
9. O. Kathib, "Real-Time Obstacle Avoidance for Robot Manipulator and Mobile Robots," *Int'l J. Robotics Research*, Vol. 5, No. 1, Spring 1986, pp. 90-98.



Tomás Lozano-Pérez is an associate professor of computer science and engineering at the Massachusetts Institute of Technology, where he is a member of the Artificial Intelligence Laboratory. His research interests are in robotics and artificial intelligence; he teaches courses in these areas at MIT. Before joining the MIT faculty in 1981 he was on the research staff at IBM's T.J. Watson Research Center during 1977 and the MIT AI Laboratory during 1974 and again during 1980. He has been co-editor of the *International Journal of Robotics Research*; co-editor of the book *Robot Motion* (MIT Press, 1982); program chairman of the 1985 IEEE International Conference on Robotics and Automation; and recipient of a 1985 National Science Foundation Presidential Young Investigator Award. He is member of the ACM and the IEEE Computer Society.

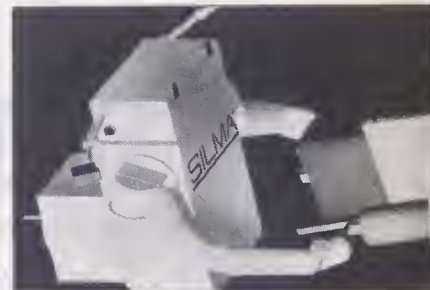
Lozano-Pérez received his BS in 1973, his MS in 1976, and his PhD in 1980, all from MIT in computer science.



Joseph L. Jones has been a research engineer at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology for six years. His interests include task-level planning and mobile robots. Previously, Jones worked as an engineering physicist at the Bates Linear Accelerator.

Jones received BS and MS degrees from MIT in 1975 and 1978, respectively.

Readers may contact Lozano-Pérez at the MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, Mass. 02139.



Emmanuel Mazer is a research fellow of the Centre National de la Recherche Scientifique (CNRS) working at the Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle (LIFIA) in Grenoble, France. His current interests are highly redundant manipulators, nanomanipulators, and automatic robot programming. Mazer has been visiting scientist at the Artificial Intelligence Laboratory of the University of Edinburgh during 1981-1982, research fellow at IMAG (Grenoble) during 1982-1984, technical director of ITMI Marketing during 1984-1986, and visiting scientist at MIT during 1986-1988.

Mazer received a master's degree in applied mathematics in 1979 and a PhD in computer science in 1988, both from Grenoble (INPG).



Patrick A. O'Donnell is currently a research engineer at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. His current interests include robotics planning, massively parallel computing, and simulation of biological structures. He is a member of ACM, the IEEE Computer Society, the IEEE Robotics and Automation Society, Tau Beta Pi, and Eta Kappa Nu.

O'Donnell received the BS and MS degrees from MIT in 1983.

MULTIPLY THE BENEFITS OF CASE

with the LAN BASED VISIBLE SOLUTION®

Coherent project development builds high quality systems faster!!!

All project team members use the same data dictionary!!!

Methodology based expert rules checks all work for consistency across the entire project!!!

Flexible, friendly, easy-to-use graphical interface brings project staff up-to-speed quickly!!!

VISIBLE ANALYST® WORKBENCH

THE NETWORKABLE VERSION 3.0--NEW FEATURES

Identical look & feel as single user version • Configurable executable modules • Unlimited number of potential users on the LAN • Little or no system degradation resulting from Workbench use • Project wide rules validation • Multiple concurrent users • Common data dictionary • Full file and record locking • Multiple security levels •

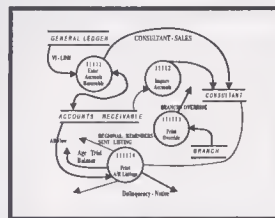
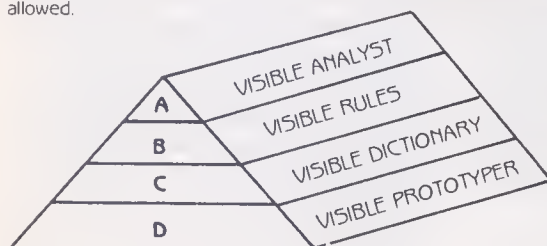
Ever since computer aided software engineering products came on the market, interest has been high. And so, not surprisingly, has been the cost of owning them. But now Visible Systems is offering state of the art CASE technology at an incredibly affordable price. \$595.00 per module. These modular tools give your company the ability to grow in CASE power. The modules of the Workbench build upon each other to provide more and more productivity.

VISIBLE ANALYST - MODULE A

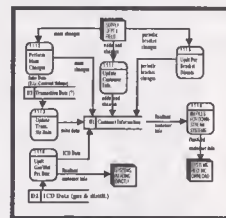
The Visible Analyst is the foundation of the Workbench. The user can create any analysis or design diagram. Multiple line styles, text fonts, and symbol types are standard features. Other features include a custom symbol generator and a reusable constructs library. Build and store entire diagrams for reuse in other projects. Boilerplates allow diagrams to be customized. Diagrams can be output in presentation quality using HP laser printers. Entire subsystems or single diagrams can be moved from one project to another while maintaining nested relationships. The LAN version allows multiple analysts to build or edit diagrams, within the same project, concurrently!!

VISIBLE RULES - MODULE B

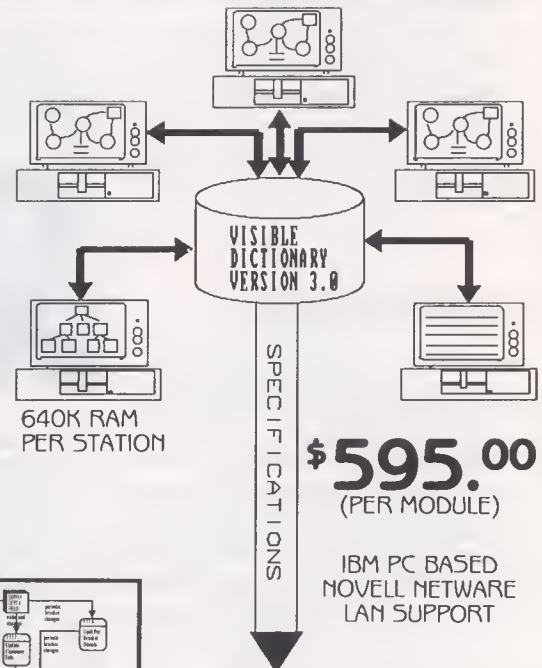
The Visible Rules is the on-line expert system that makes the use of structured techniques possible. The rules module implements either of two powerful, methodology based, expert rules systems: Yourdon/DeMarco or Gane & Sarson. Once enabled, the rules system monitors the diagramming process for naming conventions, data balancing, proper symbol usage etc. After diagrams have been built, the rules module analyzes them for errors. Error reports are output to screen or hardcopy. And the analysis takes place on-line. One never leaves the diagram. Problem resolution and identification is faster and easier. Logical data flow splitting is fully supported. Dictionary is automatically populated. Unlimited levels of nested decompositions are allowed.



YOURDON DIAGRAM



GANE & SARSON



VISIBLE DICTIONARY - MODULE C

Visible Dictionary is the central data repository for the system being modeled. It is multi-user and interactive. Many users may access it simultaneously. Full file and record locking capabilities keep users from colliding and corrupting data. Visible Dictionary has many powerful features including: "where used" listings, wild card searches and finds, full import/export, automatic entry generation and entry updating, on-line accessibility from diagrams, many different entry types, customized report formatting, and much more. The Dictionary organizes, reports, and categorizes all project data. File format information is provided for data transfers in ASCII format. The specification developed using the Visible Workbench becomes a company asset that can be used to build upon in later stages of the systems development lifecycle.

NEW!!! VISIBLE PROTOTYPING - MODULE D

The Visible Prototyper is our new screen design and full system simulation tool. Four levels of prototyping are supported: Screen design and panel linking; Panel branching, Data selection; and System simulation. User interface is "pre-approved". Integrates with all design methodologies, use for education and training-reduces costs. The Prototyper is available as a standalone or integrated fourth module with links to the Visible Dictionary for consistency in data names.

Visible Systems CORPORATION

49 Lexington Street • Newton, MA 02165 • (617) 969-4100

FAX (617) 969-9505 Telex 261102 V5CUR

LEADING EDGE CASE TECHNOLOGY

SAME DAY SHIPMENTS WORLDWIDE

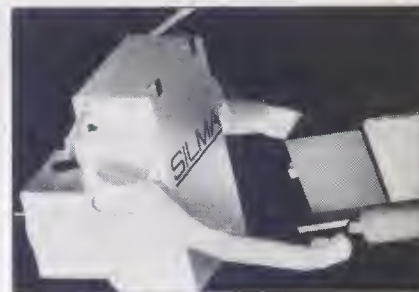
© 1988 Visible Systems Corporation

Visible, Visible Solution, The Visible Analyst and Visible Systems Corporation are registered Trademarks of Visible Systems Corporation

Reader Service Number 3

The Problems of Accuracy and Robustness in Geometric Computation

Christoph M. Hoffmann, Purdue University



Despite great advances in geometric and solid modeling, practical implementation of the various geometric operations remains error-prone, and the goal of implementing correct and robust systems for carrying out geometric computation remains elusive. This fact seems to arise from an underlying difficulty that sets geometric computation apart from other application areas in computer science and engineering.

Recently, a number of publications have begun to address this question; but there is no general agreement as to the true source of the problem or what strategies might succeed in solving it. In particular, the problem is variously characterized as a matter of achieving sufficient numerical precision, as a fundamental difficulty in dealing with interacting numeric and symbolic data, or as a problem of avoiding degenerate positions. Here, *degenerate position* could refer to positional incidence or to tangential, as opposed to transversal, intersection.

In fact, these issues are interrelated and are rooted in the problem that objects conceptually belonging to a continuous domain are analyzed by algorithms doing discrete computation, treating a very large discrete domain—for instance, the set of all representable floating-point numbers—as if it were a continuous domain. In many situations, this approach leads to

Despite the pressing need, devising accurate and robust geometric algorithms has proved elusive for many important science and engineering applications. Why is that? And, how can we change it?

acceptable results. However, except for very simple geometric objects, this simplification apparently does not work very well and expresses itself in occasional failure of the implemented algorithm operating on valid inputs.

This article examines these problems, surveys the various approaches proposed, and assesses their potential for devising complete and efficient solutions. I have

restricted my analysis to objects with linear elements, since substantial problems already arise in this case.

Linear geometric objects, in three dimensions, usually consist of points, edges, and polygonal faces in specific spatial relationship to each other. The specification of such objects consists of two parts: *numerical information*, recording (that is, vertex coordinates or plane equations), and *symbolic data* specifying face and edge boundaries, adjacencies, and incidences.

Usually, the arithmetic data describing a geometric object is given only approximately, for instance, using a floating-point representation. Consequently, the imprecise results may lead to contradictory information about the input object. For instance, the object description might require that four adjacent faces meet in a common vertex, yet the numerical plane coefficients for the faces might specify four planes that intersect in four closely spaced points rather than a single one.

In many geometric operations, the result of numerical computation must be used to infer symbolic facts. If limited-precision arithmetic is used, the outcome of the computation might crucially depend on the detailed sequence of calculations. Since uncertainty is associated with approximate arithmetic computation, some decisions are made with incomplete

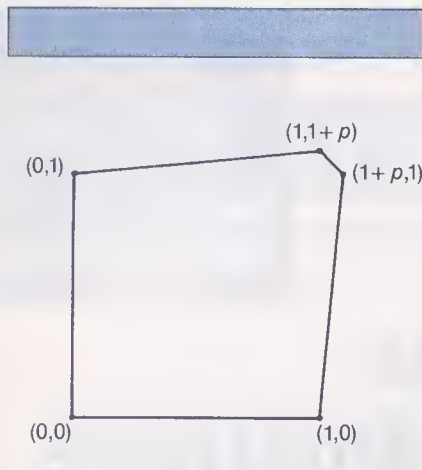


Figure 1. Example pentagon for going in and going out operations.

information. Different decisions made in this manner must be logically consistent. Moreover, certain symbolic facts can be determined by different sequences of arithmetic calculation that could result in contradictory answers. For example, determining whether a vertex u of a polyhedron A coincides with a vertex v of a polyhedron B can be done as follows:

- (1) Compute the distance to u from each of the planes intersecting in v .
- (2) If the distance from all planes is smaller than a fixed tolerance, then declare that u and v are coincident.

In this computation, interchanging the role of u and v could result in contradictory conclusions; that is, we might determine that u is incident to v but not vice versa.

Given these possibilities, we must ask the following questions: "What is meant when it is stated that a geometric algorithm is correct?" and "Does the geometric algorithm deliver an acceptable result for all legitimate inputs?" Implementing most geometric algorithms as given will lead to programs that fail for certain correct inputs.

Perturbation-free approaches

Several approaches to robust geometric computation propose to perform arithmetic with such sufficiently high accuracy that correct results must be obtained. In this section, I explore two such methods. Assuming that the input data is exact, the problem is to determine how many digits of precision the arithmetic calculations require so that the algorithm produces correct results to prescribed accuracy. I also examine whether arithmetic operations can be avoided altogether, completely replacing them with exact symbolic computation.

Floating-point computation. Using a very simple example, Dobkin and Silver illustrated the difficulty of accurately performing geometric computations repeatedly.¹ They considered a pentagon A in the plane. Drawing the five diagonals of A , their intersections define a pentagon B inside A . I call the operation of passing from A to B *going in* and denote it by $B = \text{in}(A)$. Similarly, we can extend the five sides of A to their outer intersections, thus obtaining a pentagon C containing A . I call this operation *going out*, and write $C = \text{out}(A)$. Clearly, $A = \text{out}(\text{in}(A))$ and $A = \text{in}(\text{out}(A))$. Taking a pentagon A , Dobkin and Silver iterate the going-in operation m times, obtaining $B = \text{in}^m(A)$, then compute $C = \text{out}^m(B)$. Then, they compare the coordinates of the vertices of A and C . Ideally, they should be equal. In practice, they can differ by a large error even for values of m as small as 2 or 3.

To illustrate, consider a pentagon with vertices at $(0,1)$, $(0,0)$, $(1,0)$, $(1+p,1)$, and $(1,1+p)$, for several values of m and of p (see Figure 1). Table 1 shows the results, with all computations performed in single-precision IEEE standard floating-point arithmetic. The table demonstrates dramatically that the numerical output from very simple geometric operations can be quite inaccurate.

Table 1. Absolute error for iterating going in and going out operations.

p	$\text{out}^2(\text{in}^2(\))$	$\text{out}^3(\text{in}^3(\))$	$\text{in}^2(\text{out}^2(\))$	$\text{in}^3(\text{out}^3(\))$
0.1	$3 \cdot 10^{-5}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
0.01	$2 \cdot 10^{-3}$	$9 \cdot 10^{-2}$	$7 \cdot 10^{-3}$	$7 \cdot 10^{-2}$
0.001	$2 \cdot 10^{-1}$	2	15	45
0.0001	$5 \cdot 10^{-1}$	∞	$3 \cdot 10^7$	∞
0.00001	∞	∞	$6 \cdot 10^8$	∞

Dobkin and Silver suggested computing an accuracy estimate for each geometric calculation. The inputs to the computation are systematically perturbed and the effect of the perturbation on the output is measured, essentially by repeating all computations. While they reported encouraging results, experiments with my example polygon seem less encouraging.² I will analyze why this might be.

The two operations iterated on the pentagon involve finding the equation of the line containing two given points u and v and computing the coordinates of the intersection of two lines. Of these computations, finding the intersection of two lines

$$\begin{aligned} a_1x + b_1y + c_1 &= 0 \\ a_2x + b_2y + c_2 &= 0 \end{aligned}$$

is the more delicate, as it involves inverting the matrix

$$A = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$$

The accuracy with which the matrix can be inverted can be estimated by the *condition number* κ (See Golub and van Loan³ and the sidebar entitled "Condition number.") Roughly speaking, the larger the condition number, the more the solution will be sensitive to random perturbations of the equation coefficients. Such perturbations will occur in floating-point arithmetic due to rounding and cancellation errors.

With β , the angle between two lines, we can show that the condition number of the matrix A is approximately

$$\kappa(\beta) = \frac{1}{\sin(\beta/2)}$$

Under extremely favorable circumstances, we expect a perturbation of the input coefficients of order 2^{-t} , where t is the machine precision; that is, the computed result will differ from the true result only by round-off in the last representable digit. Hence, for small intersection angles β , say $\beta = 1/2^m$, we expect to lose approximately $m+1$ binary digits.

When iterating the going-in and going-out operations, we must expect the precision loss at each stage to be multiplicative. Suppose we could guarantee an angle of no less than one arc second between any pair of lines encountered during the iteration. Then, to guarantee a meaningful result,

computing going in three times followed by three going-out operations would require doing the calculation at more than triple precision. Quadruple precision calculation would yield, in the worst case, only about four significant decimals. My experiments corroborated this finding.

Some authors have proposed specific techniques for implementing a sequence of floating-point computations in a way that minimizes errors. Such techniques, for computing the inner product of two vectors, for instance, help to minimize round-off errors; but they cannot circumvent the fact that ill-conditioned problems remain highly sensitive to minor perturbations and therefore require extended precision to reduce the relative magnitude of such perturbations.

Limited precision rational arithmetic. The previous results show that the needed arithmetic precision grows when the angle between two distinct lines diminishes. The question arises: "Are there ways to contain this growth behavior and work within a priori precision bounds?"

Sugihara proposed using exact rational arithmetic of bounded precision.⁴ He implemented Boolean operations on polyhedral solids as follows:

(1) In the representation of a polyhedron, only plane equations are given numerically, in the form $ax + by + cz + d = 0$, where a, b, c , and d are integers. All other information is symbolic: Vertices are given as the intersection of three or more planes, edges as the intersection of two planes, and so on.

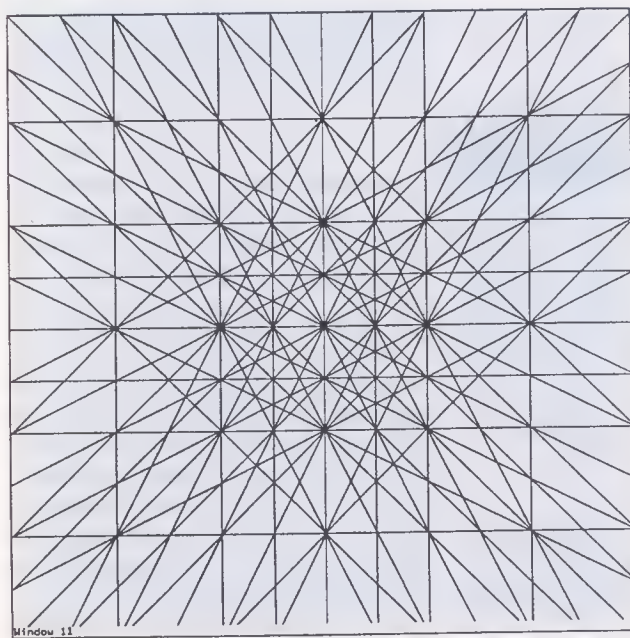
(2) A polyhedron is built from a sequence of operations on primitives, and these primitive polyhedra must be trihedral and must satisfy a minimum feature condition.

By *minimum feature condition* I mean that every edge is longer than a certain minimum and that the angle between adjacent faces is bounded away from 0 and $\pi/2$ by a minimum angle. Moreover, a trihedral polyhedron is one in which every vertex is incident to exactly three faces.

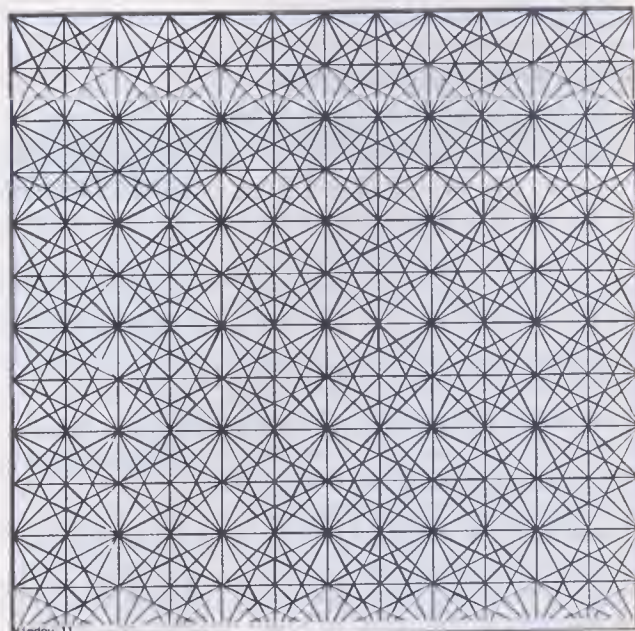
To limit the precision of the rational numbers occurring throughout the geometric computations, Sugihara required that the magnitude of a, b , and c be less than a maximum L , and that the magnitude of d be less than L^2 . The reason for giving a different bound on the magnitude of d is illustrated in Figure 2 in two dimensions. In the figure, the grid of representable lines is more uniform with $|d| < L^2$ than with $|d| < L$. The same is true in three

dimensions for the set of representable planes.

When so limiting the set of representable planes, there is clearly an absolute minimum distance between any point u that is the intersection of three planes and a fourth plane P , whenever u is not on P . This minimum distance gives rise to an estimate of the internal precision needed to correctly test incidence or nonincidence of geometric structures. If the input precision is L decimal digits, intermediate integers with up to $5L + 2$ decimal digits will be needed, since a vertex $u = P_1 \cap P_2 \cap P_3$ is on a plane P_4 precisely when the determinant



(a)



(b)

Figure 2. Grid of lines $ax + by + d = 0$, where $|a|, |b| < 3$. In (a), $|d| < 3$, and in (b) $|d| < 9$.

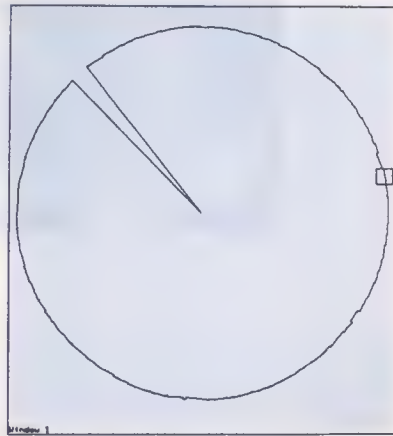


Figure 3. Union of 150 random triangles. The square indicates the area shown in Figure 4.

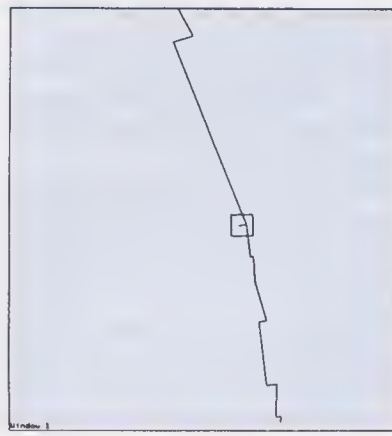


Figure 4. Border, magnified 10 times. The square indicates the area shown in Figure 5.

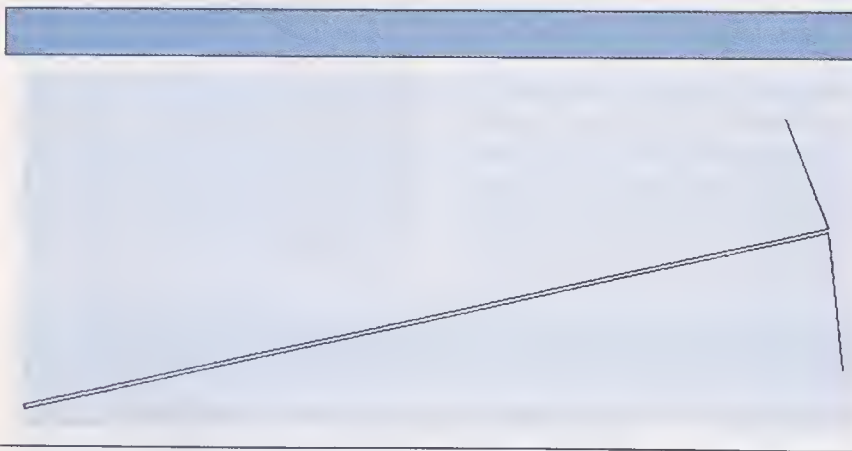


Figure 5. Border, magnified further.

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix}$$

is zero. Thus, slightly more than quintuple precision is required.

Sugihara's approach successfully deals with the problem of possible rapid growth in the number of digits needed to represent numeric data in a sequence of geometric operations. However, when using it to implement a polyhedral modeler, a drawback exists in that rotating or translating a complex polyhedron is not straightforward. To appreciate this problem, consider Figures 3-5.

Figure 3 shows the union of 150 triangles randomly generated with one vertex on a circle of radius 10^{-3} and the other two on the unit circle. Figures 4 and 5 show

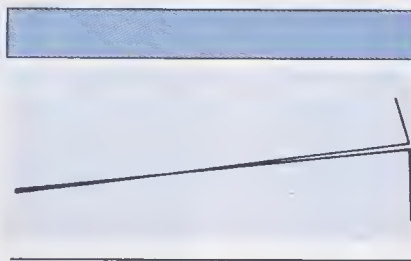


Figure 6. Possible feature alteration through translation or rotation.

details of the boundary in different magnifications. You can observe that the boundary of the resulting object contains many small features, such as the narrow crack shown in Figure 5. When translating or rotating the object, there is no guarantee that, in the new position, there will exist representable grid lines that can bound

Condition numbers

Consider inverting a matrix A . If we change the entries of A slightly, we would expect the computed inverse to change accordingly. The condition number quantifies by how much the entries of the inverse could change as a function of the changes in the matrix A . In practical terms, a condition number κ implies that changing the entries in A by an amount ϵ might result in a change in the entries of the inverse by as much as $\kappa\epsilon$. Thus, the condition number is an amplification factor for errors. It can be computed using standard techniques from numerical analysis.

When computing the inverse of A using floating-point arithmetic, we expect a result that differs from the true inverse due to round-off and other systematic errors introduced by the limited precision of floating-point arithmetic. Therefore, for a matrix A with a condition number κ , the error in the result is amplified by κ . Hence, matrices with high condition numbers are difficult to invert accurately. Such matrices are also called *ill-conditioned*.

Simplifying somewhat, I estimate that in this situation a relative perturbation of the line coefficients by ϵ leads to a relative perturbation of the intersection point coordinates proportional to $\kappa\epsilon$. Thus, κ measures how many extra digits we need in the intermediate computation so as to arrive at an inverse whose entries have the same accuracy as the line coefficients.

such a feature. In fact, on rotating or translating, each line might have to be "rounded" so its coefficients conform to the limits in magnitude. Then, the feature would possibly be altered to look as shown in Figure 6, so that a simple polygon might be changed to one that is not simple.

In view of this problem, Sugihara suggested the following: When translating or rotating a complex polyhedron P , separately translate or rotate the primitives from which P has been built, then reconstruct P from the resulting primitive objects. Since primitives are trihedral, the movement of the incident planes due to rounding might move a vertex slightly but cannot invalidate its incidence structure. For example; when rotating a cube, the faces need not remain square or parallel, although we can expect them to remain nearly so. Clearly, the possible slight

deformation of the primitives also results in a change of the topology of P . That is, the translated or rotated version of P might well differ from P in the number of vertices, faces, and edges. However, since it has been reconstructed, it must remain a valid polyhedron.

Both the need to reconstruct a polyhedron P and the fact that a move of P alters its topology are unattractive. Moreover, it is not clear how the approach can generalize to geometric objects with nonlinear elements. Two difficulties intrude: While the intersection of planes with integer coefficients is a rational point, the intersection of quadric surfaces with integer coefficients need not be rational. Already in two dimensions, the intersection of $x^2 + y^2 - 1 = 0$ and $x - y = 0$ is irrational. Moreover, while a minimum distance must exist between any two representable distinct points, estimates of the intermediate precision needed to separate them become very unfavorable. Sugihara estimates the needed internal precision for objects with quadric surfaces at 80 times the input precision.

Purely symbolic representations. The section titled "Floating-point computation" pointed out that, without further restrictions, robust geometric computation might require certain floating-point calculations to be carried out to arbitrary precision. Limited precision, on the other hand, appears cumbersome and seems difficult to generalize to nonlinear geometric elements. Could we shift the problem to symbolic computation, minimizing the role of arithmetic computation or even eliminating it?

To investigate, let us consider geometric objects consisting of lines, given as $[a, b, c]$, and points, given as (u, v, w) , where a, b, c, u, v , and w are drawn from a countable set of variables. Here, the triple $[a, b, c]$ symbolizes the line equation $ax + by + cz = 0$ and the triple (u, v, w) , projective point coordinates. Recall that the projective point (u, v, w) corresponds to the affine point $(u/w, v/w)$ whenever $w \neq 0$.

Specifying that the point $P = (u, v, w)$ is *incident* to the line $L = [a, b, c]$ shall mean that the equation $au + bv + cw = 0$ can be satisfied, and we write this fact as $L(P)$. I specify an arrangement of points and lines by the following rules:

(D1) All lines and points must be declared in advance, as triples of variables. No two lines and no two points so declared are equal.

(D2) If a point P is incident to a line L , then this fact is explicitly stated as $L(P)$. If two lines L_1 and L_2 intersect in the declared point P , then this fact is expressed explicitly by the two incidence statements $L_1(P)$ and $L_2(P)$.

(D3) No other incidences exist among declared points and lines except those explicitly stated.

These rules have been closely modeled after common conventions in boundary representations of polyhedra, where one requires that vertices, edges, and faces are all distinct, and that they do not intersect except in explicitly stated adjacencies.

Given a symbolic object specification in the methodology above, we investigate whether it can be realized as a point/line configuration in real two-dimensional projective space $\mathbf{P}^2$. That is, we ask whether an assignment of real numbers to the variables exists such that

- (1) the equations entailed by (D2) above are satisfied, and
- (2) all points and lines are distinct and satisfy (D3).

Note that this question differs from the traditional, combinatorial question whether there exists an abstract projective geometry satisfying the required conditions, since we insist on an embedding into the *specific* geometry $\mathbf{P}^2$.

It is not hard to find an example of a description that cannot be realized in $\mathbf{P}^2$: Consider the following configuration consisting of nine distinct points:

$$P_1 = (u_1, v_1, w_1), \dots, \\ P_9 = (u_9, v_9, w_9)$$

and of nine distinct lines

$$L_1 = [a_1, b_1, c_1], \dots, \\ L_9 = [a_9, b_9, c_9]$$

The required incidences are as follows:

$$L_1(P_1), L_1(P_3), L_1(P_5), L_2(P_2), L_2(P_4), \\ L_2(P_6), L_3(P_1), L_3(P_2), L_3(P_7), L_4(P_2), \\ L_4(P_3), L_4(P_9), L_5(P_3), L_5(P_4), L_5(P_8), \\ L_6(P_4), L_6(P_5), L_6(P_7), L_7(P_5), L_7(P_6), \\ L_7(P_9), L_8(P_6), L_8(P_1), L_8(P_8), L_9(P_7), \\ L_9(P_8), L_9(P_9)$$

This configuration exists in $\mathbf{P}^2$ and is shown in Figure 7. However, if the last incidence constraint, $L_9(P_9)$, is removed, then there is no such configuration in $\mathbf{P}^2$ since it contradicts Pascal's theorem. (See

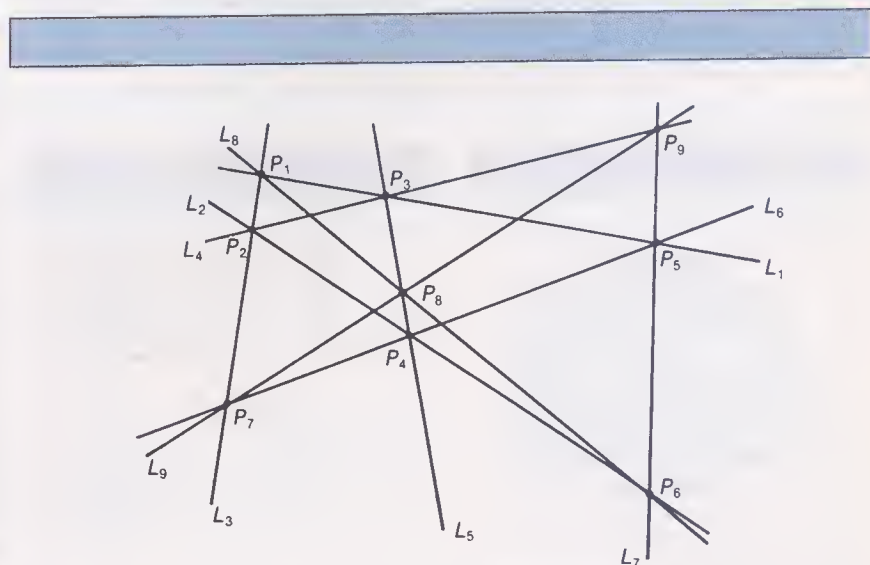


Figure 7. Realizable point/line configuration.

Pascal's theorem

In the form used here, Pascal's Theorem makes the following statement: Let $P_1, P_2, P_3, P_4, P_5, P_6$ be the vertices of a hexagon and assume that alternate vertices are on straight lines. Then, opposite sides of the hexagon intersect in three points that also lie on a straight line.

Figure 7 illustrates the theorem. The hexagon vertices P_1, P_3, P_5 lie on the line L_1 , the hexagon vertices $P_2,$

P_4, P_6 lie on the line L_2 , hence the hexagon $P_1, P_2, P_3, P_4, P_5, P_6$ satisfies the hypotheses of the theorem. P_7 is the intersection of the opposite hexagon sides (P_1, P_2) and (P_4, P_5) . Similarly, P_8 is the intersection of (P_2, P_3) and (P_5, P_6) , and P_9 is the intersection of (P_3, P_4) and (P_6, P_1) .

The conclusion is that the points lie on a common line. This line is L_3 in the figure.

the sidebar "Pascal's theorem.")

The example demonstrates that a purely symbolic representation raises logical existence problems. The solution seems to require symbolic reasoning and should lead you to ask: "How complex would the reasoning need to be?" The answer depends on the domain of geometric objects: For general geometric configurations, including curved elements, Buchberger gave an overview of several techniques.⁵ Unfortunately, these techniques have an exponential worst-case complexity. Hong observed that by choosing sufficiently long integers, the existence problem can also be settled equivalently by arithmetic computation.⁶ However, the integers needed could be of exponential length, even when restricting attention to circle and line configurations.

Later, we will see that essentially no reasoning is needed when intersecting two polygons. When intersecting three poly-

gons simultaneously, substantially more reasoning is needed, but exact complexity bounds have not yet been established. When intersecting four polygons simultaneously, we can illustrate that irrational points can be constructed from certain configurations with prescribed incidences. (Also see Hoffmann² and Hoffmann, Hopcroft, and Karasick.⁷)

Perturbation approaches

Several approaches to robust geometric computation consider altering the input data slightly. In many applications, the input data is not exact to begin with. Thus, altering it for pragmatic reasons seems appropriate. Next, let us examine several such approaches.

Representation and model. If we accept

the possibility of imprecise numerical data, we need to first explain what is described by such a representation. Only after giving a precise geometric meaning to such a representation can we decide whether a geometric algorithm has been correctly implemented. Thus, we introduce the notions of *representation* and *model*.

A representation is a description of a geometric object, possibly using imprecise arithmetic data. Such a representation has a model, if an object exists in Euclidian space satisfying the symbolic part of the description precisely whose numeric data is exact, although it might require infinite precision numbers. As we saw above, the question whether a model exists for a given representation is delicate and must not be dismissed lightly.

The definitions make a distinction between Euclidian space in which models exist and a (discrete) representation space manipulated by the computer implementation. While all geometric theorems are valid in Euclidian space, some of them might not be true for representations.

Since the numerical data in the representation is intended to be close to the exact data, comparing the exact data of the model to the approximate data of the representation makes sense. A model M of a given representation R is ϵ -close if the largest deviation of the numeric data of the representation from the exact model data is no greater than ϵ . This is an absolute error notion, which suffices for our purposes but could clearly be replaced by a relative error definition.

With these concepts, we can now clarify when a k -ary geometric operation op is correctly implemented: The implementation of op is correct if a model M_i exists for every input representation R_i such that the following is true:

- (1) The algorithm constructs an output representation R without failing.
- (2) There is a model M of R such that $M = op(M_1, \dots, M_k)$.

The definition can be further specialized to capture the precision of the algorithm as follows: Given that each model M_i is ϵ -close to its representation, the model M is $d(\epsilon)$ -close to R .

Clearly, we want assurance that a model M exists for all legitimate input data, and a function d exists such that $d(\epsilon)$ is not excessively large compared to ϵ . I will present some examples below.

The reasoning paradigm. The implementation of a geometric algorithm, operating with imprecise numeric data,

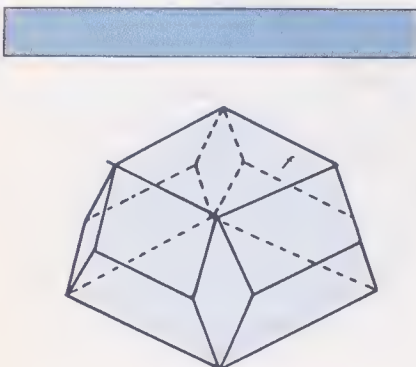


Figure 8. Repositioning the face f requires changing adjacent faces.

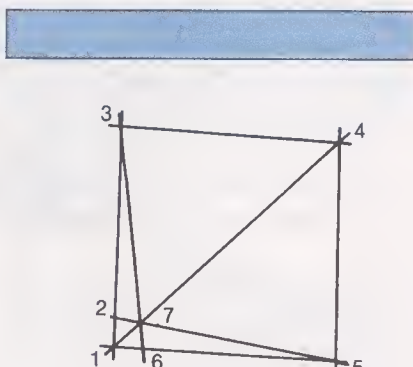


Figure 9. Sequence dependence of positional perturbation.

repeatedly involves the following scenario:

Carry out a numerical computation C . Depending on whether the outcome is positive, zero, or negative, we must make a logical decision, and that decision affects subsequent processing. As long as the magnitude of the result r exceeds a certain threshold $\tau(C)$, we can make the decision with certainty. If the magnitude of r is smaller than $\tau(C)$, the decision based on r alone is uncertain.

When the decision is uncertain, we could make it arbitrarily; that is, we could require that a result of magnitude $|r| < \tau$ is understood to mean $r = 0$. But the fact that such a decision could have consequences for other, later decisions obliges us to make each decision in a logically consistent manner. So, the reasoning paradigm says that a decision for $|r| < \tau$ requires symbolic reasoning to assure its consistency with all other such decisions.

For example, consider the problem of deciding if a vertex v of a polyhedron B is incident to a face, edge, or vertex of a polyhedron A . Suppose we have already determined that v lies in the plane containing the face f of A and that it is so close to an edge e of A that v is probably on e . If we decide that v is incident to e , then we must also decide that v is incident to the face f' of B that is adjacent to f by the edge e .

Although this example might seem trivial, it is one of the specific problems that causes many polyhedral modelers to fail for certain inputs. One of the major issues, then, is to determine how difficult—in a complexity-theoretic sense—the reasoning must be by which the algorithm establishes that the next decision is consistent with all previous decisions. Hoffmann, Hopcroft, and Karasick explored this problem.⁷ Technically, the reasoning paradigm requires the following two elements:

- (1) A specification of τ and a strategy for deciding the case $|r| < \tau$.
- (2) A proof that the program always terminates with a representation that is correct as discussed in the section titled "Representation and model."

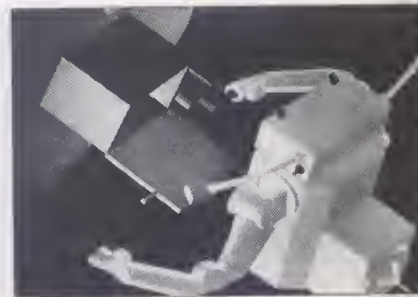
Hoffmann, Hopcroft, and Karasick showed that the intersection of two polygons can be computed in this sense with essentially no reasoning.⁷ This result follows from a method for placing edges so that all incidence constraints computed by the decision strategy for $|r| < \tau$ are satisfied. The decision strategy is simply to assume $r = 0$.

The section titled "Purely symbolic representations" showed that incidence requirements can lead to reasoning problems. Polygonal intersection is free of these difficulties as long as we do not require satisfaction of additional positional properties, such as the collinearity of nonconsecutive edges. These constraints can be introduced artificially by considering the *simultaneous* intersection of three or more polygons. For instance, the configuration of Figure 7 can be realized by superimposing three polygons.

Considerable difficulties arise for polyhedral intersection, since it is no longer evident how to reposition a face consistently so as to satisfy all incidence decisions. For example, consider the polyhedron A shown in Figure 8. Assume we need to adjust the plane containing the face f to accommodate some incidence decisions we made when analyzing the position of A with respect to some other polyhedron B . Since we must preserve the planarity of f , at most two vertices can remain in the original position. However, all vertices are adjacent to more than three faces, so that moving them also requires changing the shapes and, possibly, the positions of adjacent faces. Consequently, the operation of repositioning a face requires a *global* alteration of the polyhedron whose existence seems difficult to prove. It is therefore unclear whether the altered polyhedron representation has a model.

Let P be the plane containing a face bounded by trihedral vertices only. Then, we can alter the position of P slightly without affecting the adjacent face planes. Hence, we can prove that a trihedral polyhedron can be correctly intersected with a half space, because the discovered incidences of the vertices of the polyhedron with the intersecting plane can be satisfied by *local* positional perturbations. We cannot conclude from this that a trihedral polyhedron can be intersected with a convex polyhedron, since the result of the first intersection need not be a trihedral polyhedron. Hence, subsequent intersections might lead to incidence constraints not provably satisfiable by positional perturbations.

The stringent requirement that the topological data agree between representation and model also affects what bounds can be established on the closeness δ of the output model M as a function of the closeness ϵ of the input models M_1 and M_2 . The primary reason for this is a technical one. As stated above, when proving the correctness of an implementation in this framework,



we have to show that the elements of the input models can be consistently repositioned so as to satisfy the incidence constraints introduced during the course of the computation. In all likelihood, this repositioning is sequential, for proof purposes. But the repositioning sequence affects the final position of the vertices, edges, and faces. As an example, consider positioning the vertices 1-7 so that they satisfy the collinearity constraints depicted in Figure 9. Here, positioning them in the order 1, 3, 4, 5, 2, 7, 6 is much more favorable than positioning them in the order 1, 2, 7, 6, 3, 4, 5, since a small position perturbation of the vertices 2, 6, and 7 leads to a large perturbation of the vertices 3 and 5. Thus, not only would I like to show that a consistent sequence of repositioning operations exists for all inputs and all incidence decisions made, but the specific sequence chosen also leads to small positional perturbations.

Altering the symbolic data. So far, I have required that the symbolic data of the representation be literally satisfied by the model. Thus, the symbolic data is considered more trustworthy than the numeric data when deducing the intended meaning of a given representation. The rationale for giving priority to the symbolic data is that it can be represented easily without error. Assuming that a reliable method for defining geometric objects exists, the symbolic data in their representation ought to be correct as given.

However, objects are often constructed from other objects by geometric operations, so there is a chance that the implementation has introduced some unintended alterations into the symbolic data. Especially if some elements of an input object have been repositioned by large distances, the topology of the output object could well differ from what was intended. Therefore, if altering the sym-

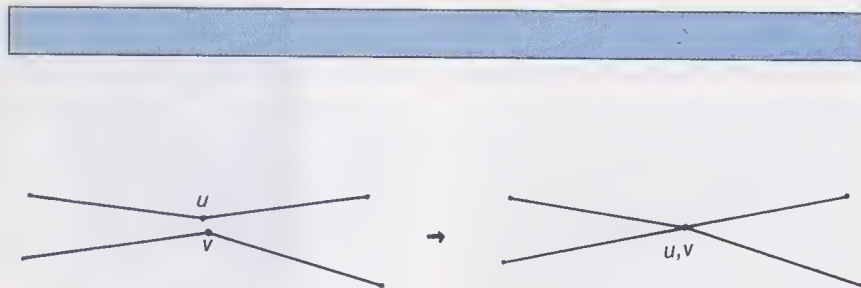


Figure 10. Vertex shifting.

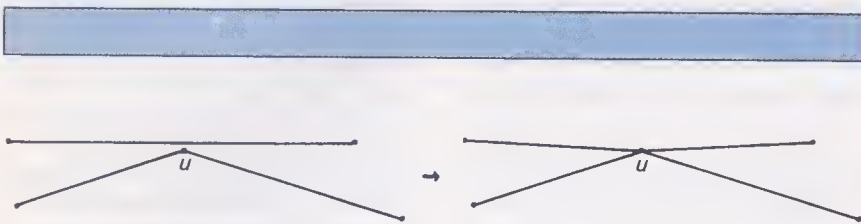


Figure 11. Edge cracking.

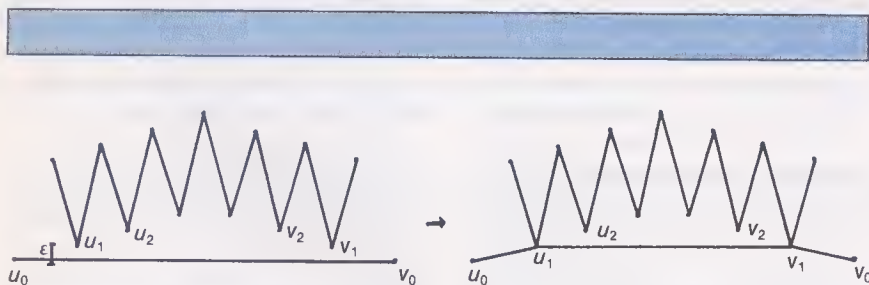


Figure 12. Additive positional perturbation in edge cracking.

bolic data slightly would result in smaller positional perturbations, I could also take the view that the numeric data is more accurate than the symbolic data. This motivates exploring the consequences of changing the symbolic data, such as, by subdividing edges and faces, followed by slight positional perturbations of the sub-

divided elements.

Milenkovic explored this approach.⁸ For operations on planar polygonal regions and for determining the topology of line arrangements in the plane, he presented two techniques:

(1) the *data normalization* method, and

(2) the *hidden variable* method.

In the data normalization method, we implement Boolean operations on polygonal regions in the plane. Each region is represented by a list of vertices whose coordinates are given as floating-point numbers and a list of edges specified by their vertices. He postulated that no two vertices are closer to each other than some tolerance ϵ and that, likewise, no vertex is closer to an edge than ϵ . The algorithm begins by altering the input data to satisfy these two requirements. Two operations, *vertex shifting* and *edge cracking*, are needed. These operations are illustrated in Figures 10 and 11.

Vertex shifting merges two vertices that are closer than ϵ into a single vertex. Since the representation is based on vertex coordinates, you can do this merging with no difficulty. Having so identified all vertices that lie close, he next subdivided any edge, provided a vertex lies close to it. If the edge is (u, v) , and w lies close to it, then (u, v) is replaced by the two edges (u, w) and (w, v) . Thus, new edges and vertices are introduced, thereby modifying the symbolic, topological data. Milenkovic proved that the algorithm terminates and gives bounds on the maximum positional perturbation.

The sequential nature of eliminating near coincidence of vertices and edges in the subdivision method can introduce positional perturbations much larger than ϵ . An example for edge cracking is shown in Figure 12. Here, the initial cracking of (u_0, v_0) by the vertices u_1 and v_1 brings the vertices u_2 and v_2 close to the middle segment (u_1, v_1) , which is cracked next. This, in turn, introduces further subdivision, so the largest displacement is proportional to $n\epsilon$, where n is the number of vertices. Milenkovic showed that, combined with errors introduced through vertex shifting, the maximum positional perturbation in the worst case is proportional to $np\epsilon$, where n is the total number of vertices, and p is the length of the perimeter.

Milenkovic proposed a second approach called the *hidden variable method* in which the positional deviation is globally bounded. The algorithm determines how line segments intersect each other within a bounding square in a topologically consistent manner. Moreover, the coordinates of all intersection points are determined to an accuracy bounded by 10η , where $\eta = 2^{-k}D$, with D the diameter of the bounding square and k the length of the mantissa. The bound is

independent of the number of lines considered.

This method replaces the input lines with *xy-monotonic curves*, that is, curves guaranteed to intersect exactly once with any line parallel to the coordinate axes. Consequently, theorems on line configurations, such as Pascal's theorem mentioned in the "Perturbation approaches" section, need not be true for the topology computed. It derived its name from the fact that these curves are never explicitly constructed but are known to exist.

Greene and Yao presented a method for drawing line segment arrangements on a discrete grid.⁹ The objective is to draw the line segments such that the end points and all segment intersections are on the grid points. To this purpose, true intersection points are rounded to the nearest grid points, and the line segments are suitably replaced with polygonal arcs.

Internally to the method, short line segments are considered between true intersection points p and their rounding p' . Such a segment is called a *hook*, and when it intersects an input line segment (u, v) , then (u, v) is also split to pass through p' . They proved that a correct topology can be constructed by first intersecting input line segments and then intersecting hooks with input line segments. Since the line segments are replaced by polygonal arcs, this approach also alters the symbolic data. However, all intermediate calculations are done with rational arithmetic, so the method is also related to those of the "Perturbation-free approaches" section.

Assuming that the grid points to which we round have integer coordinates, they proved that the internal precision needed to locate all intersections accurately depends linearly on the lengths of the coordinates of line segment endpoints.

Segal and Sequin proposed a similar method.¹⁰ Essentially, they consolidate the object representation by eliminating close incidences. The technique is similar to Milenkovic's data normalization method, but differs in that the coordinates are rounded to a grid. Small features are obliterated in this process, and the resulting objects satisfy a minimum feature condition. The algorithm assumes floating-point arithmetic.

Avoiding degeneracies. In the scenario given in the "Reasoning paradigm" section, I stated that the result of certain numeric questions is tested for its sign, and that conclusions are drawn depending on the result being positive, zero, or negative.

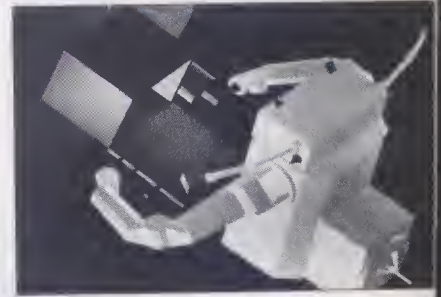
For instance, the computation might determine the Euclidean distance of a point from a line; the result would then indicate whether the point is on the positive or negative side of the line, or if it lies on the line. Let us call the latter case a positional *degeneracy*.

We can eliminate positional degeneracies by perturbing the position of these elements relative to each other, after which the scenario of the "Reasoning paradigm" section simplifies in that only two outcomes must be considered, namely, whether the result is positive or negative. The challenge in designing a perturbation scheme is to maintain the integrity of the geometric objects, whose elements are perturbed. Thus, the approach is essentially similar to the reasoning paradigm. As demonstrated in the "Purely symbolic representations" section, the difficulty in showing the correctness of a proposed perturbation scheme critically depends on the topological structure of the geometric objects manipulated.

Research has been performed proposing such perturbation schemes. The main motivation of this work is to eliminate all degeneracies. In many cases, this also simplifies the programming effort when implementing geometric operations. Note, however, that positional degeneracies are often intentional in solid modeling, and their systematic elimination would be a mistake in those applications.

For the SOS (Simulation of Simplicity) method, Edelsbrunner and Mücke proved that the final perturbations eliminate all existing degeneracies and do not create new ones not previously present.¹¹ Yap presented a similar technique.¹² Both schemes require fairly simple geometric input objects. Edelsbrunner and Mücke considered points and eliminated collinearity of three and coplanarity of four or more points.¹¹ Yap took a more abstract approach.¹² Under the assumption that all numeric computations can be expressed as fixed polynomials in the input parameters, he showed that consistent perturbations exist that eliminate the possibility of any polynomial evaluating to zero.

Results of this type would not be applicable to input objects such as polyhedra. Let A be a polyhedron input to, say, an intersection algorithm, and assume that A contains a vertex v incident to four adjacent faces that lie in the distinct planes P_1, P_2, P_3 , and P_4 . Then the topology of v implies a positional degeneracy of the four planes.



We saw in the section titled "Floating-point computation" that using unrestricted geometric computation leads to potentially unbounded growth in the number of digits that must be computed. This growth becomes especially acute when geometric operations are iterated, since the numerical errors incurred by each operation might grow multiplicatively. The growth of needed precision occurs irrespective of the topological structure of objects. However, the topology influences whether the error accumulation leads only to inaccuracy or also to internal inconsistency of the output.

Although I discussed approximate floating-point computation, digit growth can also be a problem for exact rational arithmetic unless countermeasures are built into the algorithms. It is possible to limit the growth of digits. This requires a carefully circumscribed set of permitted geometric operations. We saw in the section titled "Limited precision rational arithmetic" that useful sets of such operations exist, but the topology of the geometric objects, here polyhedra, did require expensive computations for some very simple tasks such as translating or rotating polyhedra. Moreover, it is presently unclear how to extend this approach to objects with curved boundaries.

In the "Purely symbolic representations" section, I showed that the topology of certain objects can lead to existence problems. Not all object descriptions will make sense. I concentrated on purely symbolic descriptions to show that this problem is independent of whether or not we have numeric data. However, the main conclusion to draw from the example is that the familiar description of geometric objects using approximate numeric data might contain subtle errors. To eliminate them requires potentially sophisticated



CASE '89

Third International Workshop on Computer-Aided Software Engineering

Imperial College
London, United Kingdom
July 17-21, 1989

In Cooperation With:

**The British
Computer Society**



IEEE COMPUTER SOCIETY

Call for Papers

The field of Computer-Aided Software Engineering (CASE) is the wide spread commercial application of software engineering techniques and computer technology to information systems development.

CASE '89 provides a forum for solid, detailed exchange of ideas among key practitioners, researchers, developers, and leading-edge users in the field, resulting in meaningful goals for the advancement of CASE technology.

Working groups will review the current state-of-the-art, discuss research directions, and consider future requirements in the following topic areas:

- Software Factories and Environments
 - Relationship of CASE and IPSE
 - Modelling the Software Development Process
 - Environment Architectures and Public Tool Interfaces
- Management and Social Issues
 - Technology Transfer
 - Quality and Productivity Benefits
 - Cooperative Work
 - Introduction into Practice
 - Ensuring Safe Systems; Reducing Costs
- Systems Development Methods
 - Evolution of Methods
 - Socio-technical Approaches
 - Human-Computer Interaction
- Enabling Technologies
 - Formal Methods, Prototyping, Upper-CASE Languages
 - Data, Knowledge, and Object Bases; Hypertext
 - Hardware Innovations
- Maintenance
 - Reverse Engineering
 - Reusability
- Standards for CASE

CASE '89 is a limited attendance workshop. Prospective attendees should submit 4 copies of a short position paper (500 words), in English **by April 7, 1989.**

Position papers should discuss the present state of the art, future directions of CASE technology and practical experience in the installation and use of CASE tools, or their objectives in seeking to attend CASE '89.

In particular, the Program Committee suggests the following guidelines for position papers:

- Users should discuss what capabilities are needed now and what they are seeking in the future;
- Researchers and Developers should explore what is possible to accomplish in the next five years;
- Vendors should review the present direction and constraints of CASE in the evolving five-year marketplace.

In addition, the Program Committee is seeking a number of full length papers (max. 5000 words) dealing with any of the workshop topic areas. These papers will be refereed, and we hope to invite between 20 and 30 authors to present briefly their ideas at the Workshop.

It is particularly important that all authors respect the deadline for receipt of papers.

General Chair

John O. Jenkins
Imperial College
School of Management
London, SW7 2PG, United Kingdom
(+44) 01.589.5111 ext. 7112
fax: (+44) 01.584.7596

Program Chair

M. M. Lehman
Lehman Software Technology
Associates and Imperial College

North American Coordinator

Elliot J. Chikofsky
Index Technology Corporation
One Main Street
Cambridge, MA 02142 USA
617-494-8200 ext. 1989
chikofsky@northeastern.edu

Program Committee

M. Lehman (UK)
C. Batini (I)
B. Bouldin (US)
R. Carasik (US)
E. Chikofsky (US)
A. Cowderoy (UK)
J. C. Derniame (F)
K. Ehrlich (US)
J. Hawgood (UK)
J. Jenkins (UK)
J. Lambert (UK)
T. Maibaum (UK)
M. Morron (UK)
D. Sutherland (US)
G. Tate (NZ)
R. Verbruggen (Ire.)
J. Verner (NZ)
M. Verrall (UK)
M. Vine (UK)
J. v. Zijl (Neth.)

Sponsored by:

Imperial College

In Cooperation With:
British Computer Society
Computer Society of the IEEE
Index Technology Corporation
PA Consultancy
Imperial Software Technology
The Sema Group plc
Washington University CSDP

and expensive symbolic computation.

The existence problem is one motivation for drawing a distinction between representation and model, as shown in the "Representation and model" section. We cannot naively assume that a given representation makes sense even though, based on approximate metric data, the computer is able to give, for instance, a graphical rendering of it. A second motivation for the reasoning paradigm is that, in many applications, the numerical data cannot be known precisely either because it is arrived at through approximate computation or because it is based on noisy measurement. Therefore, we seem to be left with the following choices:

- (1) Guarantee exact data, as in the "Limited precision rational arithmetic" section.
- (2) Include reasoning steps into the computation, as in the "Representa-

tation and model" and "Reasoning paradigm" sections.

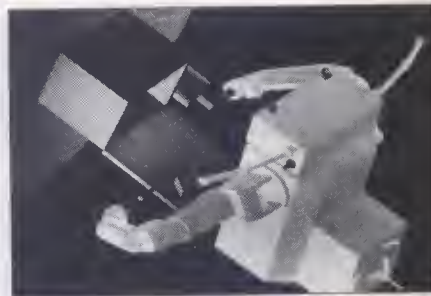
- (3) Alter the meaning of geometric elements, as in the "Altering the symbolic data" and "Avoiding degeneracies" sections.

It is not presently possible to make clear recommendations as to which alternative is preferable. More research is needed to develop the various approaches before the ultimate merits of each idea can be fairly judged for each application area. □

Acknowledgments

I am pleased to acknowledge the long and fruitful collaboration with John Hopcroft that led to the approach sketched in the "Representation and model" and "Reasoning paradigm" sections. In the solid modeler of Mike Karasick,¹³ I experimented with the ideas presented in those two sections of this article. Figures 2-5

were inspired by similar figures in Sugihara⁴ and were made with a program implemented by Ching-Shoei Chiang and Jia-Xun Yu. My research was supported in part by National Science Foundation Grant CCR 86-19817, Office of Naval Research Contract N00014-86-K-0465, and a grant from the AT&T Foundation.



References

1. D. Dobkin and D. Silver, "Recipes for Geometry and Numerical Analysis, Part 1: An Empirical Study," *Proc. Fourth ACM Symp. on Computer Geometry*, June 1988, pp. 93-105.
2. C. Hoffmann, *Geometric and Solid Modeling*, Morgan Kaufman, San Francisco, to be published in 1989, chapter 4.
3. G. Golub and C. van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, chapter 2.
4. K. Sugihara, "An Approach to Error-Free Solid Modeling," *IMA Summer Program on Robotics*, Inst. Math. and Applications, Univ. of Minnesota, Minneapolis, Minn., 1987.
5. B. Buchberger, G. Collins, and B. Kutzler, "Algebraic Methods for Geometric Reasoning," *Ann. Reviews in Computer Science*, Ann. Reviews, Inc., Palo Alto, Calif., Vol. 3, 1988, pp. 85-119.
6. J. Hong, "Proving by Example and Gap Theorems," *Proc. 27th Symp. Foundations of Computer Science*, November 1986, pp. 107-116.
7. C. Hoffmann, J. Hopcroft, and M. Karasick, "Towards Implementing Robust Geometric Computations," *Proc. Fourth ACM Symp. on Computer Geometry*, June 1988, pp. 106-117.
8. V. Milenkovic, "Verifiable Implementations of Geometric Algorithms Using Finite Precision Arithmetic," Tech. Report CS-88-168, Carnegie Mellon Univ., Pittsburgh, 1988.
9. D. Greene and F. Yao, "Finite Resolution Computational Geometry," *Proc. 27th IEEE Symp. Foundations of Computer Science*, November 1986, pp. 143-152.
10. M. Segal and C. Sequin, "Consistent Calculations for Solids Modeling," *Artificial Intelligence*, D. Kapur and J. Mundy, eds., special issue on geometric reasoning, Elsevier, North-Holland, Vol. 37, 1988.
11. H. Edelsbrunner and E. Mücke, "Simulation of Simplicity: A Technique to Cope with Degenerate Cases in Geometric Algorithms," *Proc. Fourth ACM Symp. on Computer Geometry*, June 1988, pp. 118-133.
12. C. Yap, "A Geometric Consistency Theorem for a Symbolic Perturbation Theorem," *Proc. Fourth ACM Symp. on Computer Geometry*, June 1988, pp. 134-142.
13. C. Hoffmann, J. Hopcroft, and M. Karasick, "Robust Set Operations on Polyhedral Solids," Tech. Report 723, Computer Science Dept., Purdue University, W. Lafayette, Ind., 1988.



Christoph M. Hoffmann, who is completing authorship of a book on geometric and solid modeling, has held faculty positions at the University of Waterloo in Canada and Purdue University in Indiana, and visiting faculty positions at the University of Kiel in West Germany and Cornell University in New York. He has done research in programming languages and compiler design, algorithms for computational algebra and graph theory, and solid modeling and robotics.

Hoffmann received the BS degree in mathematics from the University of Hamburg in West Germany, the MS degree in mathematics from Indiana University, and the PhD degree in computer science in 1974 from the University of Wisconsin.

Readers may write to Hoffmann at the Computer Science Dept., Purdue University, West Lafayette, IN 47907.

CONFERENCE ON RESEARCH OPPORTUNITIES FOR BREAKTHROUGH INITIATIVES

SOFTWARE TECHNOLOGY FOR ADAPTABLE RELIABLE SYSTEMS (STARS)

Atlanta, GA. March 20, 1989

SOFTWARE ENGINEERING TECHNOLOGY, INC. on behalf of the Defense Advanced Research Projects Agency (DARPA) Software Technology for Adaptable Reliable Systems (STARS) Program announces a technical briefing and conference to identify the areas with research "breakthrough" opportunities to reduce overall risk in the STARS system development process. Currently Boeing, IBM, and UNISYS have initiated a five year ADA software development environment incorporating a software first life cycle. This conference and industry briefing presents some of the opportunities for breakthrough initiative research projects.

The conference and industry briefing is designed to stimulate proposals for innovative "breakthrough initiatives" from the software research community in areas such as:

SOFTWARE FIRST LIFE CYCLE
ADA ENVIRONMENT
SOFTWARE QUALITY AND RELIABILITY
PROTOTYPING AND INCREMENTAL DEVELOPMENT
REPOSITORY OPERATIONS
REUSE METHODS

STARS '89 BREAKTHROUGH INITIATIVES CONFERENCE

DATE: March 20, 1989 9:00 A.M.

PLACE: Atlanta Airport Marriott
4711 Best Road
Atlanta, GA 30337

ROOM RESERVATIONS: (404)766-7900 OR (800)228-9290

PROGRAM CHAIR AND PANEL

CHAIR: Dr. Harlan D. Mills, Information Systems Institute

PANEL: Dr. Alan R. Hevner, University of Maryland
Dr. Jesse H. Poore, University of Tennessee
Dr. Stephen M. Thebaut, University of Florida
Dr. Marvin Zerkowitz, University of Maryland

REGISTRATION AND INQUIRIES

Arnold R. Beckhardt
Software Engineering Technology, Inc.
2770 Indian River Blvd.
Vero Beach, FL 32960
Phone: (407) 569-6644; fax (407) 569-9303

Exception Handling in Robotics

Ingemar J. Cox and Narain H. Gehani
AT&T Bell Laboratories



A robot program can be logically correct yet fail under abnormal conditions. A major goal of robotics research is to construct robust and reliable robot systems able to handle errors arising from abnormal operating conditions. Consequently, error handling and recovery is becoming increasingly important as researchers strive to construct reliable, autonomous robot systems for factory, space, underwater, and hazardous environments.

Although commercial robot programming languages continue to be special-purpose languages, researchers increasingly agree that general-purpose programming languages are sufficient, and indeed preferable, for robot programming.^{1,2} This is because, in a typical robot application, robot-specific operations constitute a small fraction of the total program. The remainder typically concerns operator interfaces, database access, communication, synchronization, numerical computation, and error handling.

Advances in computer science research in language design have strongly influenced robot programming. For example, the advantages of data abstraction, such as program modularization and information hiding, have received widespread attention within the robotics community.^{3,4} Communication and synchronization have particular importance for robot programming. In fact, it has been observed that, in

Errors caused by abnormal external events can cause logically correct robot programs to fail. Exception handling facilities provide a mechanism for writing reliable, robust, and maintainable robot programs.

the design of a program involving two cooperating robots, up to 30 percent of the code concerned the interaction between the two robots. Although conventional general-purpose languages do not adequately support these requirements, general-purpose languages with appropriate concurrent programming facilities do meet them.²

In this article we discuss the error handling technique called exception handling

as it applies to robotics. Exception handling facilities help build reliable robot systems that, under many circumstances, can recover from previous error states. We describe the type of errors that can occur in a robot system and their error handling requirements and review approaches to error handling and recovery. We mention the exception handling facilities in Ada and AML/X¹ and briefly describe Exceptional C,⁵ the language we chose for exposition. Finally, we illustrate the use of exception handling facilities with two examples. We assume that the reader is familiar with C.

Errors and error handling requirements

We can classify robot system failures into three categories: software errors, hardware errors, and external state errors. We don't address software errors or programming bugs here. Rather, we assume that the robot program is logically correct, but fails due to hardware or external state errors.

Hardware errors describe the physical failure of robot subsystems, such as a sensor system. It might or might not be possible to recover from hardware errors. In the worst case, the failure of a nonredundant robot subsystem, such as a motor actuator, must result in the failure of the robot program. The best you can hope for is that the system can achieve a safe termination state.

Other hardware failures might be less catastrophic, such as the failure of a single sensor subsystem. In many cases, the application might be able to continue to function by using an alternative sensor. For example, an ultrasonic transducer might provide an inexpensive substitute for an optical rangefinder, albeit with poorer performance. In addition, the receipt of erroneous data from a sensor might result from a temporary failure of the sensor. In that case, you can recover from the error by simply rereading the sensor.

Designing fault-tolerant systems offers one solution to hardware error. Key subsystems are replicated and work in parallel. If one of the replicated components fails, the remaining components continue to operate.⁶ The user does not notice the error, and the program continues to execute correctly. However, fault tolerance is orthogonal to our theme here; we are not interested in the design of fault-tolerant systems that mask errors by using redundancy. Instead, given that errors will occur, we want to know how to handle them. (Note that fault-tolerant systems cannot mask all types of errors and that even fault-tolerant systems can fail, albeit with less frequency.)

External state errors describe circumstances where the robot's assumed state of the world does not match that perceived by its sensors. For example, an autonomous robot vehicle might unexpectedly find an obstacle in its path. Alternatively, a robot might find that it has failed to grasp a required object either because the object is not present or because the object slipped from its grip. These are the types of error conditions that intelligent robots must be designed to handle and recover from. We cannot predict the occurrence of such errors, and we can only prevent them if we have tight control over the robot's environment. In many situations, particularly in space, underwater, and hazardous environments, this is simply not possible.

Requirements for robot error handling will vary with the application. However, a common requirement is the need to recover from an error in real time, where the real-time constraint depends upon the application. A second characteristic, which supports the real-time issue, is that error recovery should be (loosely) hierarchical. For example, the presence of an obstacle in the path of an autonomous vehicle need not immediately invoke the path planner to construct a new route, avoiding the obstacle. Rather, it seems perfectly reasonable for the vehicle to first

wait some amount of time in the hope that the obstacle will move. Only if this simple error recovery procedure fails should it be necessary to propagate the error up to a higher level of software.

Two distinct approaches deal with errors in robotics. The first approach handles "errors" that are part of the normal execution of a robot program. For example, robot manipulators have positional and repeatability errors, so the dimensions of the parts that the robot manipulates are accurate only to the specified design tolerances. The known uncertainties in part sizes and robot/part position allow you to determine a priori that a particular assembly task has a high likelihood of failure. Given this knowledge, you must modify the robot action to accommodate for uncertainties present in the system. You would usually accomplish this through sensors.

Compliant motion, which provides a good example of this approach, has received considerable attention in the robotics community. Compliant motion allows modification of the robot's trajectory during the motion by force data obtained from sensors placed on the robot arm.

At a somewhat higher level is the concept of a plan checker.⁷ Here, a robot planner decomposes a task into a series of robot actions. Given information on the robot's positional accuracy, the tolerance of the parts, etc., the plan checker determines whether the set of actions will succeed. Any action considered likely to fail results in modification of the plan in an attempt to ensure success.

The first approach, although extremely important, involves the development of robust programs that deal with routinely present "noise" in the system. The second approach deals with abnormal conditions that arise during the execution of the program, for example, recovering from the situation when a part falls from a gripper.

The second approach to robot error handling complements the first. Specifically, we are interested in the use of special facilities provided in some programming languages to handle abnormal conditions that arise in robot programs.

Most work on handling abnormal conditions in robots has centered on task-planning approaches,⁸ not on programming language support facilities. Programming language facilities for error handling provide support for the detection and notification of abnormal conditions and subsequent invocation of the error handling code, be it calls to a planner or other appropriate code.

Programming language facilities

We can classify error handling facilities into two categories: backward and forward error recovery. In the backward error recovery mechanisms, upon the detection of an error, execution is rolled back to a checkpoint and resumed from there. One backward error recovery mechanism, the *recovery block*,⁹ is expressed linguistically as

```
ensure acceptance-test by algorithm1
else by algorithm2
...
else by algorithmn
```

Execution of the Ensure statement occurs as follows: a checkpoint of the program state is taken at the beginning, then *algorithm₁* is executed. After this, the acceptance-test expression is evaluated. If true, this completes the execution of the Ensure statement. Otherwise, the program is rolled back to the checkpoint established at the beginning and *algorithm₂* is tried. If none of the algorithms *algorithm₁*, ..., *algorithm_{n-1}* satisfy the acceptance test, then *algorithm_n* is executed.

In forward error recovery mechanisms, of which exception handling facilities are a prime example, the occurrence of an error transfers control to the error handling part of the program. After the error has been handled, execution continues from the point specified in the exception handler.

Backward error handling facilities might not in general suit robotic applications because in many situations it might not be appropriate for the robot to return to a previous state and try again. For example, if a hole has not been drilled properly, then going back certainly cannot imply undrilling the hole. Consequently, we will focus on the forward error recovery mechanism called exception handling.

Exception handling

An *exception* is an event that occurs infrequently, possibly indicating an error, such as division by zero. As programs grow larger, a standard mechanism for handling exceptions becomes important. An *exception mechanism* is a language control structure that allows a programmer to express that the standard continuation of a program is to be replaced by an exceptional continuation when an exception is detected in the program.¹⁰ An *exception handling mechanism* is a tool for enhancing program reliability and fault tolerance.

In languages without exception handling facilities, we must use other methods to indicate exceptions and to handle exceptions. The most common method is to have functions return status values or "funny values" to indicate exceptional conditions. If the value returned by a function indicates that an exception has occurred during the function call, then appropriate action is taken. The status value technique

- (1) leads to difficult-to-read programs in which it is not immediately obvious that errors are being checked for,
- (2) makes programs inefficient because a check is made to determine the function status every time it is called, even though the error occurs rarely, and
- (3) increases the size and complexity of the programs because of the checks that must be inserted.

In an effort to reduce programming complexity, programmers often omit error handling code in the initial stages of program development with the intention of adding it later. Unfortunately, this code might never be put in; even if it is put in, it might have a negative impact on the program structure.

The status value technique does not allow the detection and handling of processor exceptions, such as one raised by an attempt to divide by zero, nor can this technique help detect and handle exceptions raised from the program environment, such as an exception raised by a user wanting to terminate program execution prematurely. The status value technique must be supplemented by some other mechanism, such as a signaling mechanism supported by the operating system.

Exception handling facilities typically do not have the problems associated with the status value technique.

There are two exception handling models^{11,12}:

(1) *The termination model*: The program component in which an exception occurs is terminated after the exception has been handled and execution is resumed in its caller. This model is used in the Ada and CLU languages.

(2) *The resumption model*: In this model, after the exception has been handled (and any error situation remedied), the handler can cause the program component in which the exception occurred to be retried, resumed (at the point where the exception occurred), continued (with the next program component), or terminated. This model is used in PL/I and Mesa.

The termination model, although simple to understand and implement, is not as expressive as the resumption model.

Exception handling facilities in different languages

A few languages, such as PL/I and Ada, provide exception handling facilities. PL/I was the first major language to do so. AML/X, an important robot programming language, and Exceptional C,⁵ an experimental superset of C, both provide exception handling facilities.

Ada. Ada uses the termination model of exception handling. Exceptions must be declared, and exception handlers are associated with blocks. Raising an exception results in suspension of execution of the normal part of a program. The statements specified in an exception handler are then executed to handle the exception.

Exception handlers are specified at the end of a block, a subprogram, a package, or a task. Execution of an exception handler completes execution of the block, subprogram, package, or task. If no exception handlers are provided, then execution of the program segment in which the exception was raised is abandoned, and the responsibility of responding to the exception is transferred to another part of the program.

The specific exception handler activated to take care of an exception depends upon the execution path of the program, not its textual layout — the association of a handler with an exception is dynamic, not static.

AML/X. AML/X supports exception handling by means of library routines.¹ It provides a function to bind the exception handler to an exception. Exception handlers are not restricted to be just statements; for example, an exception handler can be an expression (which becomes the result of the operation raising the exception) or a statement label (control is transferred to the specified label when the exception is raised).

Exceptional C. Exceptional C supports a hybrid termination-resumption model. Exceptions are classified into two categories: An ordinary exception normally indicates an error in the program. A signal exception does not indicate a fault in the program, but is used for interprogram or

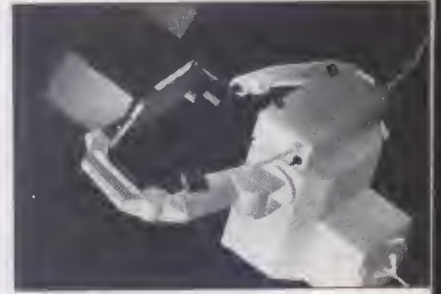
environment-to-program communication. An example of an ordinary exception is a floating-point overflow; an example of a signal exception is an interrupt from a mouse.

Exception handlers are associated with blocks (that is, compound statements; note that function bodies in C are compound statements). In case of an ordinary exception, statements in the exception handler can possibly fix the error raising the exception; then the block containing the statement raising the exception can be reexecuted. Alternatively, execution of the block containing the statement can be terminated and the exception, optionally, propagated to the caller of the block.

Since a signal exception does not indicate a program error, the program execution can resume at the point of interruption (the point where the signal exception was raised) and the interrupted block can be reexecuted. Reexecuting a statement (to be precise, the block containing it), instead of resuming its execution, might be necessary if interruption of a statement requires that it be executed again to get correct results. An example of such a case is an interrupted I/O statement. If a block does not contain a handler for a signal exception, then a handler is searched for up along the dynamic call chain for the block. If no exception handler can be found, then the program terminates.

Local retries and continues are allowed for all exceptions. Resumption semantics are allowed for signal exceptions because there is nothing really wrong with the program itself.

Exceptions must be declared either by using an exception declaration or in a function header. The designer of a function specifies the exceptions that can be raised in a function and propagated to its caller. Information supplied in the function declaration is used to check whether handlers are provided for the specified exceptions



(warnings are issued to the user) and, if one is supplied, whether or not its parameter types match. Other than the exceptions specified in a function header, a function can propagate only the exception Error to its caller.

The Raise statement is used to explicitly raise exceptions.

Exception handlers are specified by means of an Except statement, given as the last statement in a block. The Except statement has the form

```
except {
    when  $e_{11}$  | ... |  $e_{1n}$  : statements1
    when  $e_{21}$  | ... |  $e_{2n}$  : statements2
    ...
    when  $e_{m1}$  | ... |  $e_{mn}$  : statementsm
    when others: statementsm+1
}
```

where e_{ij} is an exception name (optionally followed by parameters).

When an exception is raised in a block, control jumps to the handler for the exception associated with the block. If there is no handler for the exception, but there is the catchall exception Others, then control transfers to the catchall handler. Otherwise, the block terminates and the exception is propagated to the surrounding block or, in the absence of a surrounding block, to the caller of the function.

When a signal exception is raised during the execution of a block, control jumps to the handler associated with the block. If one is not found, then one is searched for in the surrounding block or, in case there is no surrounding block, to the caller of the function. Note that unlike the case of an ordinary exception, the block where the exception is raised is not terminated nor is the Others clause used to handle a signal exception.

Execution of an exception handler is terminated by completing the statements in the handler or by exiting from the handler by executing the Return statement or by executing one of the following exception handling extensions: Retry, Next, Resume, or Raise statements.

The block containing the statement where an exception is raised can be reexecuted by executing the Retry statement, which has the form

```
retry;
```

The Retry facility might superficially resemble a backward error recovery mechanism, like the recovery block, but there are important differences: there is no

checkpointing of the state, and you first go forward to the exception handler, where you can attempt to fix the problem before going back to Retry.

You can specify the number of times a block is retried by using the form

```
retry(number-of-times);
```

Exceptional C keeps track of the Retry count (associated with the handler containing the Retry statement). This counter resets to zero every time the block containing the exception handler is entered afresh.

The statement following the block in which an exception is raised can be executed by executing the Next statement, which has the form

```
next;
```

This statement is useful for continuing with the next iteration of a loop.

Programs can be resumed at the point they were interrupted because of a signal exception by executing the Resume statement:

```
resume;
```

To help debug a program, Exceptional C provides facilities for printing the name of the file containing the statement that raised the exception and the source file line number.

Examples

In this section, we present two examples illustrating the use of exception handling in the context of an autonomous vehicle.³ Note that, at the time of writing, these examples had not been implemented on the vehicle but had been run under simulated conditions using Exceptional C.

Autonomous vehicle. In this example, a robot vehicle is given the desired destination, consisting of an x , y position and an orientation. A call to a path planner generates a path consisting of a series of line and arc segments. While moving along the path, the vehicle might come across an obstacle. In many circumstances, the obstacle is transient — simply a person crossing the vehicle's path. However, occasionally, furniture or other objects present a permanent obstruction to the vehicle. In these situations, the cart's map of its environment is updated to include the new obstacle, and a new path is generated.

We now show the program that handles

errors involving obstacles encountered by the robot vehicle. First, Figure 1 shows the file containing the declarations. Figure 2 shows the Main program. The Main program reads in a desired destination and calls the path planner, which constructs a series of maneuvers needed to accomplish the motion. The path planner returns the total number of segments in the path. The program then enters a loop (line 24) calling the Move routine to advance the vehicle along each of the path segments. The Move routine is synchronous, returning only after the motion has been completed.

If the cart encounters an obstacle, then exception Obstacle is raised and control jumps to the exception handler (lines 8-13). The handler contains code to wait for the specified time period and retry the move (in the hope that the path of the cart was obstructed by a transient obstacle). The code specifies a maximum of three retries. If even after the three retries the cart cannot move to the specified destination, then the exception handler in Move raises the exception Cannot_move. This exception is propagated to the Main function and handled by its exception handler (lines 27-33). This exception handler updates the map, inserting the current obstacle into the database, and retries the block (starting with line 23, which calls the path planner). Given the updated map, the path planner constructs a new path to avoid the obstacle. If after three tries (each one done after a map update) the cart cannot avoid the obstacle, the program terminates with an appropriate message.

Autonomous vehicle with redundant rangefinders. Consider an autonomous vehicle equipped with two rangefinders: an optical rangefinder and an ultrasonic rangefinder. The optical rangefinder scans a beam in a plane via a rotating mirror. After each complete scan, which takes about one second, a zero-reset interrupt is generated. Assume that a watchdog timer

```
#define TIME 3 /*seconds*/
#define XMAX 512
#define YMAX 512
#define MAX 16

5 void wait(), updateMap();
  int pathPlanner();
  short map[XMAX][YMAX];
```

Figure 1. File cart.h.

indicates a scan error if such an interrupt is not received within two seconds. The ultrasonic rangefinder, consisting of a ring of ultrasonic sensors, serves as a backup. Because of its much poorer performance characteristics, it is used only in cases where the optical rangefinder malfunctions, such as when the mechanical scanning mechanism fails.

By checking the signal strength detected

by either sensor, it is possible to determine the validity of the computed range value. If the signal strength is too weak (insufficient signal) or too strong (the detector is saturated), a bad range value is indicated.

Figures 3 and 4 show the program that handles errors involving the two rangefinders. Figure 3 is file range.h, which contains the appropriate declarations. Figure 4 is the Main program.



```
#include "cart.h"
#include <stdio.h>
void move(x, y, theta) raises(exception CANNOT_MOVE);
float x, y, theta; /*theta is the orientation*/
5 {
    exception OBSTACLE;
    move the cart
    except {
        when OBSTACLE:
10         wait(TIME);
            retry(3);
            raise CANNOT_MOVE;
        }
    }
15 main()
{
    float x, y, theta, X[MAX], Y[MAX], Theta[MAX]; int n, i;

    initialize map and associate the obstacle exception with the
    rangefinder
;

20 /*read and execute moves*/
while (scanf("%f%f%f", &x, &y, &theta) == 3)
{
    n = pathPlanner(x, y, theta, X, Y, Theta, MAX);
    for (i=0; i<n; i++)
25     move(X[i], Y[i], Theta[i]);
    except {
        when CANNOT_MOVE:
            updateMap();
            fprintf(stderr, "going to retry!\n");
            retry(3);
30             fprintf(stderr, "Cannot reach destination!\n");
                exit(1);
        }
    }
35 }
void updateMap()
{ update map }

int pathPlanner(x, y, theta, X, Y, Theta, n)
float x, y, theta, X[], Y[], Theta[];
40 int n;
{
    compute new path and return number of segments
}
void wait(t)
45 int t;
{ wait t seconds }
```

Figure 2. Main program, file cart.ce.

```
double (*rangefinder)() raises(exception
    BAD_RANGE);
double ultrasonic_rf() raises(exception BAD_RANGE);
double optical_rf() raises(exception BAD_RANGE);
exception BAD_RANGE;
5 signal exception SCAN_ERROR;
```

Figure 3. File range.h.

```
#include <stdio.h>
#include "range.h"
main()
{
5   double r[72]; int angle;
    signal exception SCAN_ERROR;
    associate the scan error interrupt with the signal exception
    SCAN_ERROR;
    rangefinder = optical_rf;
    for (angle = 0; angle < 360; angle += 5)
10    {
        r[angle/5] = (*rangefinder)(angle);
        ...
        except {
            when BAD_RANGE:
15             printf("bad data\n");
                retry(2);
                if (rangefinder == optical_rf)
                    rangefinder = ultrasonic_rf;
            else {
                printf("quitting!\n"); exit(0);
            }
            when SCAN_ERROR:
20             rangefinder = ultrasonic_rf;
                printf("scan error, switching to ultrasonic!\n");
                resume;
        }
25    }
}
```

Figure 4. Main program, file cart2.ce.

```

#include <stdio.h>
#include "range.h"
double optical_rf(angle) raises(exception BAD_RANGE;)
    int angle;
5 {
    determine range
    if (range value is bad)
        raise BAD_RANGE;
    return range
10 except {
    when BAD_RANGE: raise BAD_RANGE;
    when SCAN_ERROR:
        rangefinder = ultrasonic_rf;
        printf("scan error, switching to ultrasonic!\n");
15         raise BAD_RANGE;
    }
}
double ultrasonic_rf(angle) raises(exception BAD_RANGE;)
    int angle;
20 {
    determine range
    if (range value is bad)
        raise BAD_RANGE;
    return range
25 except {
    when BAD_RANGE: raise BAD_RANGE;
    }
}

```

Figure 5. File range.ce.

Association of an interrupt with a signal exception that is not predefined is implementation dependent (line 7). Initially, the optical rangefinder is assumed operational and the function pointer Rangefinder is assigned the address of the optical rangefinder (line 8). The rangefinder is called on line 11. When the rangefinder functions raise the Bad_range exception, control transfers to the exception handling code (lines 13-26). The handler for Bad_range retries the rangefinder twice (code starting from line 10 is reexecuted). If the exception Bad_range is raised by each retry, then the program switches from the optical rangefinder to the ultrasonic rangefinder. If this has already been done, the program terminates (lines 17-21).

The signal exception Scan_error is handled differently depending on whether it is raised within the function Optical_rf. If it is raised outside this function, control jumps to the exception handling code (line 21). The handler for Scan_error simply reconfigures the program to use the ultrasonic rangefinder (line 23) and resumes execution from the point of interruption.

Figure 5 shows the code for the range-

SENIOR COMPUTING SCIENTIST

Reporting to a Vice President/Technical Director, will help to conceive, develop and apply state of the art computing technologies in diverse technical areas such as artificial intelligence, image processing, graphics and parallel computing. Will be a key contributor to the direction of TASC's highly-regarded Computing Technology Center and will assume major responsibility advising our Computing Resources Steering Group. Will also be responsible for some activities involving marketing, proposals and business development related to new research and applications for advanced computing technology. The right person will have a Ph.D. in Computer Science, 10 years' related experience, a solid blend of academic, industrial and/or research and development experience, and should be published and known in the computing sciences world.

Please write to Coley I. Rybicki

An Equal Opportunity
Employer, M/F.
U.S. Citizenship
Required.

TASC
THE ANALYTIC SCIENCES CORPORATION

55 Walkers Brook Drive
Reading, MA 01867

CALL FOR PAPERS

The 3rd International SOFTWARE FOR STRATEGIC SYSTEMS CONFERENCE

HUNTSVILLE, ALABAMA ♦ FEBRUARY 27-28, 1990

This international conference will build upon the results of the first two conferences held in May 1987 and Oct. 1988. Special emphasis will be placed on advanced technological concepts needed for strategic software systems. The conference provides a unique opportunity to participate in an international forum emphasizing software issues of strategic importance.

Unclassified papers are sought in the following areas:

Algorithms	Man/Machine Interface
Artificial Intelligence	Modeling & Simulation
Automated Decisions Aids	Software Environments & Tools
Computer Security	Software Dev. Methodologies
Data Base Concepts	Software Reliability
Fault Tolerance	Software Testing/Testbeds
Knowledge-Based Applications	

Deadlines & Requirements:

5/31/89 - Submit in English 4 draft copies of an unclassified paper not to exceed 20 double-spaced, typewritten pages. Include the full name, title, affiliation, address & phone number of each author.
7/31/89 - Notification of acceptance ♦ 10/31/89 - Camera-ready copy due

Submit papers or inquiries to:

The University of Alabama in Huntsville, Division of Continuing Education, Tom Beville Center 285, Huntsville, Alabama 35899 or call (205) 895-6372 or toll-free 1-800-448-4035 outside Huntsville.



The University
Of Alabama
In Huntsville

An Affirmative Action/Equal Opportunity Institution

finder functions.

If the optical rangefinder (function `Optical_rf`) detects a bad range, it raises the exception `Bad_range` (as specified in its header on line 3). This exception is raised explicitly on line 8. The ultrasonic rangefinder does the same thing for bad ranges. If the signal exception `Scan_error` is raised during the execution of the function `Optical_rf`, then it is handled locally (lines 10-16). Note that both exception handlers propagate the exception `Bad_range` to the caller.

Error handling and recovery are becoming increasingly important for robot applications, especially for autonomous operation in underwater, space, and hazardous environments. Exception handling facilities, such as those provided in Exceptional C, provide a mechanism for writing easier to read, reliable, and robust robot programs. The programs become easier to maintain because normal program code is separate and distinct from the error handling code. This makes it easier to understand the algorithm controlling the robot in both normal and abnormal conditions. The propagation of exceptions up the function (or block) call chain provides a hierarchical structure to error handling. At the lowest level, simple error recovery strategies such as wait and retry can first be attempted before propagating the error to higher and correspondingly more sophisticated levels of exception handling.

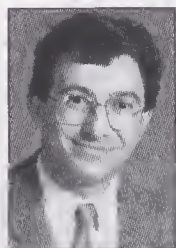
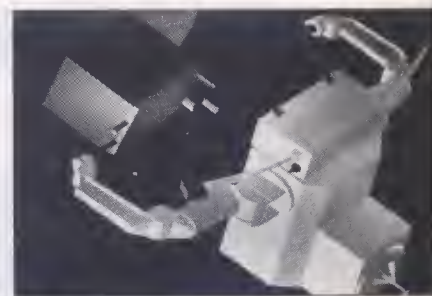
The use of exception handling facilities in the examples given above significantly simplifies error handling. Moreover, the error handling code is clearly distinct from the normal code. Without the exception handling facilities, the code for the normal and error cases would intermingle. Finally, we should point out that in a language that does not support exception handling it is not straightforward to simulate this functionality, especially in the context of exception propagation and flow of control. □

Acknowledgment

We are grateful to the anonymous referees and Gordon T. Wilfong for their many helpful comments.

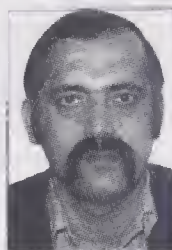
References

1. L.R. Nackman et al., "AML/X: A Programming Language for Design and Manufacturing," *Proc. 1986 Fall Joint Computer Conf.*, 1986, pp. 145-159.
2. I.J. Cox and N. Gehani, "Concurrent Programming and Robotics," *J. Robotics Research* (to be published).
3. I.J. Cox, "Blanche: An Autonomous Robot Vehicle for Structured Environments," *IEEE Int'l Conf. Robotics and Automation*, 1988, pp. 978-982.
4. R.A. Volz and T.N. Mudge, "Robots are (Nothing More Than) Abstract Data Types," *Proc. SME Conf. Robotics Research*, Aug. 1984.
5. N.H. Gehani, "Exceptional C," AT&T Bell Laboratories (to be published).
6. R.F. Cmelik, N.H. Gehani, and W.D. Roome, "Fault-Tolerant Concurrent C: A Tool for Writing Fault-Tolerant Distributed Programs," *Proc. 18th Int'l Symp. Fault-Tolerant Computing*, June 1988, Tokyo, Japan, pp. 56-61.
7. R.A. Brooks, "Symbolic Error Analysis and Robot Planning," *J. Robotics Research*, Vol. 1, No. 4, 1982, pp. 29-68.
8. E.D. Sacerdoti, "Planning in a Hierarchy of
9. B. Randell, "System Structure for Fault Tolerance," *Proc. Int'l Conf. Reliable Software, SIGPLAN Notices*, Vol. 10, No. 6, June 1975, pp. 437-449.
10. F. Cristian, "Exception Handling," *IBM Research Report 5724*, 1987.
11. J.B. Goodenough, "Exception Handling: Issues and a Proposed Notation," *Comm. ACM*, Vol. 18, No. 12, 1975, pp. 683-696.
12. R.A. Levin, *Program Structures for Exceptional Conditional Handling*, PhD thesis, Carnegie Mellon University, 1977.



Ingemar J. Cox has been with AT&T Bell Laboratories since 1984. His current interests are broadly in the areas of autonomous robot vehicles, computer vision, and real-time programming environments.

Cox received a BS from London University in 1980 and a PhD from Oxford University in 1983.



Narain H. Gehani has been with AT&T Bell Laboratories since 1978. His research interests include concurrent programming, programming methodology, programming language design, office automation formal specifications, compilers, and databases. He has been involved in the design and development of Concurrent C, a tool for distributed programming. His current work involves the design and development of Exceptional C, a tool for writing reliable programs, and ODE, an object-oriented database.

Gehani is currently an ACM National Lecturer. He received his PhD in computer science from Cornell University in 1975.

Readers may contact the authors at AT&T Bell Laboratories, Room 2T-406, 600 Mountain Ave., Murray Hill, NJ 07974.

Current Research in Robotics and Automation

Four overviews provide a sampling of current research in robotics. Included are reports from Columbia University, Sandia National Laboratories, Cornell University, and the University of Southern California.

An Intelligent Grasping System

Peter Allen, Paul Michelman, and Kenneth Roberts

Department of Computer Science, Columbia University, New York, NY 10027

Work in dextrous manipulation at Columbia University's Computer Science Department focuses on building a comprehensive grasping environment capable of performing tasks such as locating moving objects and picking them up, manipulating man-made objects such as tools, and recognizing unknown objects through touch. In addition, we are designing an integrated programming environment that will allow grasping and grasping primitives within an overall robotic control and programming system that includes dextrous hands, vision sensors, and multiple-degree-of-freedom manipulators.

Our current system contains the following components:

- a set of low-level system primitives that serve as the basis for the control of the hand,
- a true hand/arm system with many degrees of freedom formed by mounting the hand on a robotic manipulator, a Puma 560,
- integrated tactile sensors on the fingertips that provide force-sensitive responses and Cartesian position information, and
- sensing primitives that use joint position, tendon force, and tactile array

sensing in a number of grasping and manipulation tasks.

System overview. The system we have built consists of a Utah-MIT hand¹ attached to a Puma 560 manipulator (Figure 1). The hand has four fingers, each with four degrees of freedom. It resembles the human hand in size and shape, but it lacks a number of features humans find very useful. In particular, it has no palmar degree of freedom (closing of the palm), and the thumb is placed directly opposite the other three fingers, with all fingers identical in size. The hand has joint-position sensors that yield joint-angle data and tendon-force sensors that measure forces on each of the two tendons (extensor and flexor) controlling a joint. The Puma adds six degrees of freedom to the system (three translation parameters to move the hand in space and three rotational parameters to orient the hand), yielding a 22-degree-of-freedom system. The system is controlled by an intelligent, high-level controller that links arm, hand, and finger movements with the feedback sensing of joint positions, tendon forces, and tactile responses on the fingers.

Figure 2 shows the system's hardware structure. The high-level control resides in

a Sun-3 processor. The Sun serves as the central controller and has access to a full Unix-based system for program development and debugging as well as a set of window-based utilities to allow graphical output and display of the system's various states. The hand is controlled by an analog controller commanded through D/A boards from a dedicated 68020 system. The Sun is capable of downloading and executing code on the 68020 and can communicate with it through a shared memory interface.² The low-level system primitives to control the hand include the ability to read and write joint positions and tendon forces, and forward and inverse kinematics to allow both joint-space and Cartesian-space operation of the hand. The joint-position and tendon-force sensors can also be monitored at high sampling rates to monitor the hand's operation in real time, for both movement and contact grasping.

The tactile sensing system is controlled by another dedicated 68020 that monitors the forces on each of the sensor pads. The tactile sensors we are using consist of 16×16 grids of piezoresistive polymeric material that are conformable to the finger's shape.³ They are manufactured by sandwiching the polymeric material

between two pliable sheets of Kapton that contain electrical etching. The application of forces on the pads provides an increased electrical flow channel between the two sheets as the material within is compressed. Results with this sensor have been good, particularly with respect to signal isolation. The sensors are monitored by a separate 68020 that is responsible for low-level tactile processing, including A/D conversion, thresholding, and signal normalization. A number of tactile filters, including edge and line detection, smoothing, and moment analysis, have also been implemented.

The control of the Puma arm is through a serial link that allows the Sun to act as the Puma system host (host control option). This allows the system to use the full range of robotic arm commands. These include movement primitives, asynchronous interrupt capability, and the ability to establish

arbitrary coordinate frames such as the hand coordinate system. It also provides a global coordinate system in which the tasks can take place.

Applications. The system has performed a number of grasping tasks, including pick and place operations, extraction of circuit boards from card cages, pouring of liquids from pitchers, and removing light bulbs from sockets. These tasks have been programmed using Dial, a graphical animation language developed by Steven Feiner.⁴ This diagrammatic language allows representation of parallel processes in a compact graphical "time line" and has been used for animation of graphical displays. We transported it to the robotics domain and have exploited its ability to express parallel operation of robotic devices. Dial also provides a convenient way to implement



Figure 1. Utah/MIT hand mounted on Puma 560.

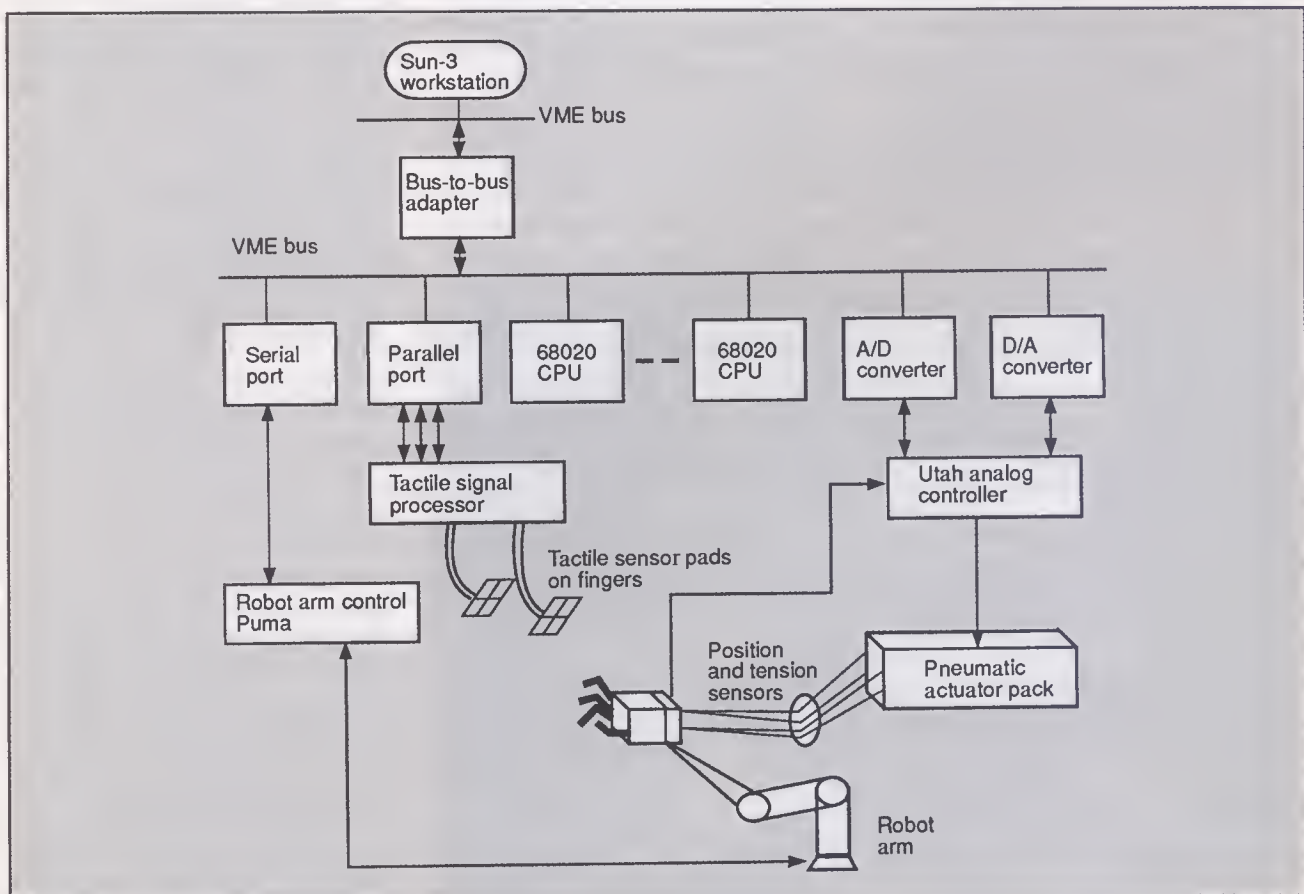


Figure 2. Hardware overview.

task-level scripts, which can then be bound to particular sensors, actuators, and methods for accomplishing a generic grasping or manipulation task. The instruction set Dial supports is specified at runtime in a user-supplied back end.

The task of removing a light bulb from a socket requires tendon-force feedback to determine the grasp strength and the coordination of the fingers moving in parallel to unscrew the bulb. The task is performed by moving the arm to the light bulb while preshaping the hand, and then repeating the following sequence of events: grasp the light bulb, turn the hand/arm counter-clockwise, release the grasp, and move back to the initial preshaped position by rotating the hand and arm clockwise. After the bulb becomes loose, the hand grasps the bulb and retracts it from the socket.

A second application is haptic object recognition — that is, understanding shape from touch. It is becoming clear that robotic systems need capabilities similar to the human haptic system in order to perform complex grasping, manipulation, and object-recognition tasks using dextrous hands. Research in psychology has shown the ability of humans to accurately discriminate an object's shape by touch alone.⁵ We have attempted to emulate this human ability by recovering the shape of objects through grasping. We have chosen to model objects as superquadrics. These models are derived from a parameterization that allows a wide degree of freedom in modeling objects. The parameter space is continuous and allows a smooth change from a cuboid to a sphere to a cylinder, with more complex shapes derivable with the addition of bending and tapering parameters. These "lumps of clay" are deformable by the usual linear stretching and scaling operations and can be combined using Boolean set operations to create more complex objects.

In addition to possessing most of the desirable modeling criteria discussed above, the models are surprisingly easy to recover from sparse and noisy sensor data. The small number of parameters needed to recover the shape describes a very large class of objects that can be equated with generic or prototypical recognition. Efficient recovery of superellipsoids (a sub-

class of superquadrics) from vision and range data has already been shown in the work of Pentland,⁶ Solina,⁷ and Boulton.⁸

What makes superquadrics particularly relevant for haptic recognition is the following:

- The models are volumetric in nature, which maps directly into the psychophysical perception processes suggested by grasping by containment.
- The models can be constrained by the volumetric constraint implied by the joint positions on each finger.
- The models can be recovered with sparse amounts of point contact data, since only a limited number of parameters need to be recovered. There are five parameters related to shape and six related to position and orientation in space. Global deformations (tapering, bending) may add a few more.
- In addition to the use of contact points of fingers on a surface, the surface normals from contacts can be used to describe a dual superquadric that has the same analytical properties as the model itself.

In this procedure, the Puma arm moves the hand to a position in which it will close around the object. Then the fingers are closed by position commands until the observed force (estimated by the difference between the tendon forces) exceeds a given threshold, which indicates that the finger is in contact with the object. The joint-angle positions are read, and kinematic models of the hand and the Puma arm are used to convert them to XYZ positions in world coordinates. This sparse set of points is then injected into a recovery algorithm developed by Solina⁷ that uses a nonlinear estimation technique to recover the superquadric parameters.

We tested this procedure against a database of six objects, including objects that could be modeled as undeformed superquadrics (block, large cylinder, small cylinder) and deformed superquadrics (light bulb, funnel, triangular wedge).⁹ Each object's shape was recovered accurately with extremely sparse amounts of data, typically 30-100 points. Note that this is about two orders of magnitude less than typical range data images, which try to recover shape with denser data that, unlike touch sensing, is limited to a viewpoint

that only exposes half the object's surfaces to the sensor. We are currently extending this method to include more exploratory procedures for understanding shape from touch. □

Acknowledgments

This work was supported in part by DARPA contract N00039-84-C-0165, and NSF grants DMC-86-05065, DCI-86-08845, CCR-86-2709, and IRI-86-57151. Thanks also to Steven Feiner and Cliff Beshers for bringing Dial to our attention.

References

1. S.C. Jacobsen et al., "Design of the Utah/MIT Dextrous Hand," *Proc. 1986 IEEE Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., order number 695 (microfiche only), pp. 1,520-1,532.
2. S. Narasimhan, D.M. Siegel, and J.M. Hollerbach, "Condor: A Revised Architecture for Controlling the Utah-MIT Hand," *Proc. 1988 IEEE Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., 3-vol. set, order number 852, pp. 446-449.
3. *The Force Sensing Resistor: A New Tool in Sensor Technology*, Interlink, Santa Barbara, Calif., 1986.
4. S. Feiner, D. Salesin, and T. Banchoff, "Dial: A Diagrammatic Animation Language," *IEEE Computer Graphics and Applications*, Vol. 2, No. 7, Sept. 1982, pp. 43-54.
5. S. Lederman and R. Klatzky, "Hand Movements: A Window into Haptic Object Recognition," *Cognitive Psychology*, Vol. 19, 1987, pp. 342-368.
6. A.P. Pentland, "Recognition by Parts," Tech. Report 406, SR1 Int'l, Menlo Park, Calif., 1986.
7. F. Solina, *Shape Recovery and Segmentation with Deformable Part Models*, PhD dissertation, Dept. of Computer Science, Univ. of Pennsylvania, Philadelphia, Dec. 1987.
8. T. Boulton and A. Gross, "Recovery of Superquadrics from Depth Information," *Proc. AAAI Workshop Spatial-Reasoning and Multisensor Integration*, Morgan Kaufman, Los Altos, Calif., Oct. 1987.
9. P. Allen and K.S. Roberts, "Haptic Object Recognition Using a Multifingered Dextrous Hand," Tech. Report CUCS-363-88, Dept. of Computer Science, Columbia Univ., New York, Aug. 1988.

Automated Planning and Programming for Robotic Batch Mechanical Assembly

Patrick J. Eicker and David R. Strip
Sandia National Laboratories, Albuquerque, NM 87185

As a prime contractor to the US Department of Energy, Sandia has a strong commitment to developing advanced robotic and other automation systems.

Batch and hazardous operations. Certain Sandia applications have operations where the presence of humans is denied or restricted. In some cases, the operations are hazardous because of the presence of ionizing radiation or explosive materials. For instance, robotic systems are being considered for use in a proposed geologic repository for commercial nuclear waste as a way of limiting radiation exposure of repository personnel. Robots may also provide the additional benefit of excellent throughput in the repository. In other cases, the presence of humans poses hazards to the success of operations. Sandia operates a Class 1 clean room for producing integrated circuits. Here, human operators are being replaced with ultraclean automatic equipment by using mobile robots that have a top-mounted pedestal robot to handle material.

Batch manufacturing is another robotics application of importance to Sandia. We design special devices that are produced within the DoE manufacturing complex. At most, a few thousand items are built over the production lifetime. Most of these are classical industrial electronic and electromechanical devices, which may be distinguished by special tolerance and reliability requirements. Figure 1 shows an assembly that exemplifies a family of electromechanical switches designed at Sandia. Certain of these devices contain explosive or radioactive components. A number of the production processes have been targeted for automation, ranging from machining and finish-

ing of piece parts to assembly of sub-systems.

Automating such production processes can improve safety, quality, and productivity. However, traditional industrial robotics and automation are not applicable in many cases. The difficulty in applying traditional systems is twofold: First, in batch and hazardous operations, teaching robots is expensive, difficult, or impossible; second, fixturing is expensive (in the case of batch operations) or difficult (as would be the case with a hundred-ton radioactive waste cask).

To overcome the difficulties of programming robots, we seek automatic programming methods; to achieve this we use various forms of computer-based reasoning. This overview outlines our algorithm and software-related research and devel-

opment for automated mechanical assembly. A more complete discussion of Sandia's robotics work is given in Eicker.¹

Automated mechanical assembly. Automated planning and programming is arguably the most important technology to future robotic systems but is perhaps the least well-developed. At Sandia our applications span the classical gamut of reasoning requirements — motion planning, assembly planning, grasp planning, manipulation planning, object identification, object categorization, and so on. Over the last decade and a half, the international community has carried out a significant amount of research in these areas; however, it is mostly conceptual and works poorly, if at all, when applied to real devices and problems.

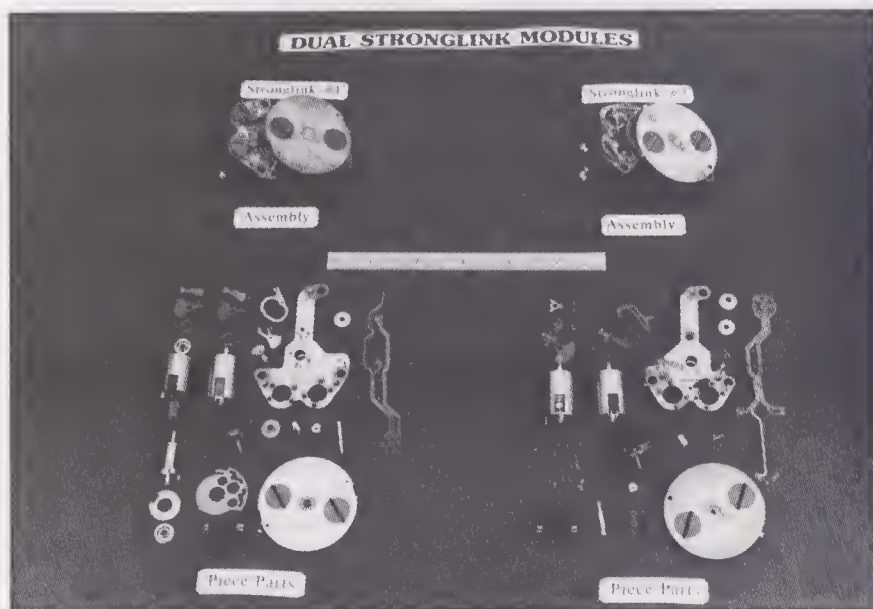


Figure 1. Piece parts and subassembly of Sandia's electromechanical lock.



Figure 2. Test setup for insert primitive.

Robotic mechanical assembly typifies the requirements for new technology. In contrast to electronics, where robots are beginning to penetrate the assembly market, mechanical assembly is not well supported by today's robotic systems. A succinct explanation of this difference is difficult; suffice it to say that mechanical assemblies have more degrees of freedom than electronic, implying a greater complexity in motion planning and execution. Further, the shapes and relations of the parts being assembled in mechanical systems are more directly related to the device's overall function than in electronic assembly, implying that the assembly system designer can have little influence on device tolerances. A variety of new algorithmic and software techniques are needed if flexible automation, such as robots, is to be useful in the future for batch mechanical manufacturing.

Research areas. At Sandia, there are five major areas of ongoing research:

- compliant motion,
- motion planning,
- machine interpretation of solid models,
- automatic planning of assembly sequences, and
- automatic programming of robotic systems.

Compliant motion primitives are necessary to minimize the detail with which the robot programmer must deal. Key primitives for the assembly process include

insert; align to face, edge, or corner; and follow contour. Insertion is the most important of these, and we have developed a single primitive which works for all convex-shaped objects.² This insertion program has been demonstrated to work with unchamfered objects having clearances less than 0.001 inches in robotic systems whose accuracy is only 0.1 inches, using only programmed compliances. A test setup is shown in Figure 2. Use of insert is simple; it requires only low-accuracy knowledge of the initial and final location of the part as well as a simply defined target point on the part. All details of the insertion are handled by the low-level robot controller.

The motion-planning problem has been studied for a number of years. Much of this research has concentrated on automatically generating a trajectory that moves the end effector from one position to another in such a way that the arm does not encounter any obstacles in the workspace. We have carried out work in this area in several programs in which we have used wire frame models of the workpiece and robot to generate safe approach paths to the workpiece.^{3,4} In assembly planning, cell design, which is the inverse of motion planning, is of more importance and interest. Using the solid model of the assembly and its piece parts as a starting point, our research aims to automatically lay out a work cell in which all motions will be collision-free and in which workspace and assembly time are optimized. Furthermore, the motion planner must automatically discover and link the appropriate compliant motions.

The long-term vision for mechanical assembly is a completely automated operation in which "electronic drawings" for a device are received and, without human intervention, an assembled mechanism is delivered. An important issue in our research is the content the input drawings must have to support the automated process. Our current research takes the solid model of the assembly as a starting point and aims to understand the extensions to solid modeling necessary to support this process. For example, the following concerns are not yet addressed in solid models, but related information must be available to the assembly system.

(1) Solid modelers do not deal with parts in contact, but transient contacts will occur as the parts are being assembled.

(2) Solid modelers do not yet provide for most of the textual information, such as tolerances, contained in engineering drawings, but such information is critical for assembly.

(3) Solid modelers do not provide information on the order in which parts must be assembled, but correct order is necessary in many cases—for example, to keep the device together in intermediate stages of assembly. The assembly planner must be able to use the input model of the device, and the model of the assembly system, to automatically generate the order plan for assembly.

Finally, the system must be able to automatically generate the detailed program for assembly. To a degree, this can be thought of as an assembly compiler that uses the assembly plans to generate instructions for the particular mechanical, sensing, and computing systems. We have implemented such an automatic linker for several systems,⁴ but much additional research is needed for the assembly area. A key research topic will be the automatic generation of execution monitoring and error detection/correction programs. Ultimately, this work will result in a hierarchy of "languages" that describe the assembly operations and generate the correct system responses. □

References

1. P.J. Eicker, "Robotics Research at Sandia National Laboratories," *Proc. Int'l Symp. Robots*, Springer-Verlag, New York, 1988, pp. 1051-1066.
2. D.R. Strip, "Insertions Using Geometric Analysis and Hybrid Force-Position Control: Method and Analysis," *Proc. 1988 IEEE Int'l Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., 3-vol. set, order number 852, pp. 1,744-1,751.
3. D.J. Miller, "Supervisory Control for a Complex Robotic System," Tech. Paper MS88-358, Society of Manufacturing Engineers, Dearborn, Mich., 1988.
4. C.S. Louchk, C.B. Selleck, S. Thunborg, "CAD-Directed Robotic Edge Finishing," Tech. Paper MR89-114, Society of Manufacturing Engineers, Dearborn, Mich.

Electronic Prototyping

John E. Hopcroft

Department of Computer Science, 4130 Upson Hall, Cornell University, Ithaca, NY 14853

Electronic prototyping consists of building a computer model of an object to verify its design. Traditionally, manufacturers have built physical prototypes of complex objects to verify design criteria — for example, to verify that no two parts occupy the same physical volume and that each serviceable component has a free path for removal during servicing. Today, we are on the verge of having sufficient computing power to replace physical prototyping with electronic prototyping. However, a number of obstacles are barring progress. Cornell's Modeling and Simulation Project was created to develop the necessary science base to overcome these barriers. Here we describe the project's progress in improving robustness of solid modelers and its ongoing research in electronic prototyping.

Solid modeler robustness. Computer-aided design systems have created a need for robust geometrical algorithms. Existing algorithms are known to fail on certain inputs for numerical reasons. The cause of these failures usually can be traced to logical decisions such as "branch-on-zero" that are based on the results of numerical calculations. Numerical inaccuracies, introduced either in the initial data or in the finite-precision arithmetic used, may result in a set of logical decisions that are inconsistent. If the algorithm's correctness depends on the logical consistency of the underlying structures, the loss of these properties usually proves fatal to the algorithm.

Our research in geometric modeling resulted in a paradigm that should have wide applicability for producing provably correct programs for engineering applications such as solid modeling. The representation of most geometrical objects consists of two types of data: symbolic and numeric. In the case of a polyhedron, the symbolic data consists of names of vertices, edges, and faces, along with edge-vertex and face-edge incident relations. Calculations involving only symbolic data are logical in nature. Since symbolic data does not degrade with computation, a program involving only symbolic data should not fail unless there is a flaw in the program itself.

The numerical data usually consists of approximations to the real values of various entities. In the case of a polyhedron, it consists of the coefficients of the face equations represented to some precision. A program that performs only numeric computations will fail only if a division by zero occurs. An argument based on continuity can be used to establish that the result is an approximation to the correct answer. However, conversions of numeric data to symbolic data within a computation create difficulties. Typically, such a conversion takes place when the result of a numeric computation is used in a branch-on-zero operation. If the numerical quantity is very close to zero, one must decide if the quantity is actually zero or if the difference is round-off error. If only one such conversion from approximate numeric data to symbolic data occurs, it is not likely to lead to a difficulty, since a

slight perturbation of the numeric input data could force the result of the computation to be either zero or not zero. Thus, no inconsistency is likely to arise from either decision. However, if a sequence of such conversions are made, the results may not be logically independent in the sense that each possible set of answers can arise from some set of input values. Once the symbolic data is inconsistent, some property that is essential for the correctness of the algorithm may fail to hold. Indeed, this is the difficulty that causes many implementations of geometrical algorithms to fail. The solution to this problem is to structure the algorithm, if possible, so that all conversions are logically independent.

This paradigm has been used to develop an intersection algorithm that is provably correct, even with numerical round-off. When employed for the intersection of general polyhedral objects, the algorithm is several orders of magnitude more robust than existing codes. An early implementation by Karasick¹ has shown that one can take a cube, rotate it an angle θ about each of its axes, and intersect the cube with its rotated version. Existing commercial modelers tend to crash for angles of θ in the range of two degrees to one-half degree. We have successfully intersected objects with rotated versions down to $\theta = 1/10,000$ degree. At $\theta = \frac{1}{2}$ degree the result is the actual intersection. Below $1/10,000$ degree the algorithm concludes that the two objects are identical and returns the original object as the intersection. Between these values the algorithm

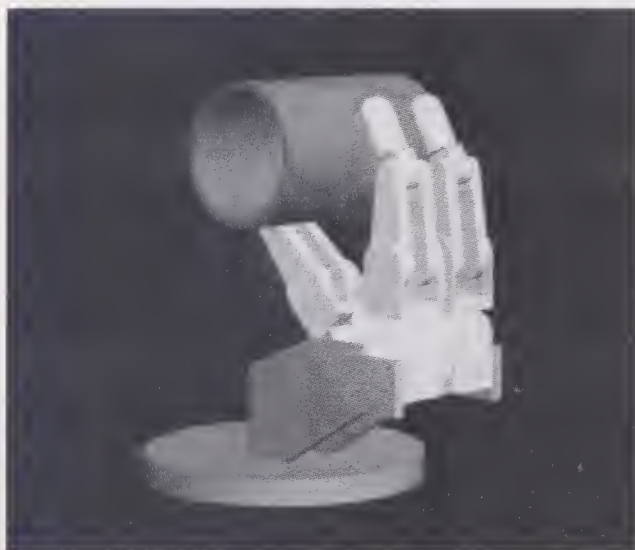


Figure 1. Electronic prototypes, like this one of the Stanford JPL gripper, are easier to modify than physical models.

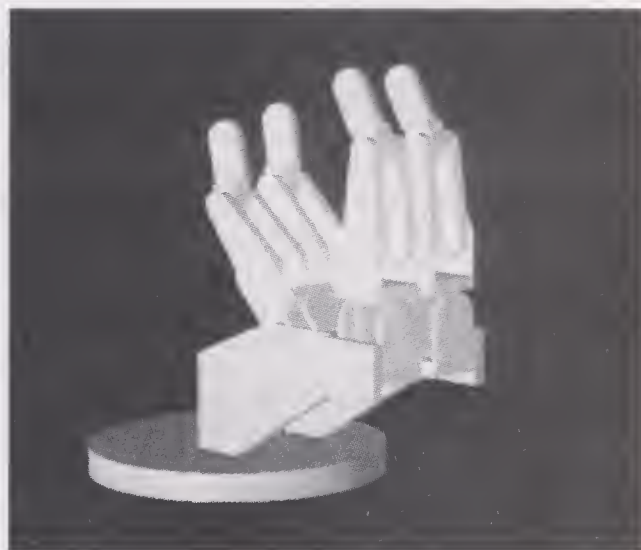


Figure 2. Parameters used in Figure 1 changed to display a four-fingered version of the Stanford JPL gripper.

gracefully converges from one behavior to the other.

Electronic prototyping. Part of our research effort concerns electronic prototyping. Such prototypes have several advantages over their more traditional physical counterparts. First, designs can be changed much later in the design/test cycle, and the system can be revalidated by rerunning a validation program. Furthermore, electronic models can be used to check many aspects of the system not available to physical models.

One of our first electronic prototypes was of the Stanford JPL gripper, shown in Figure 1. It can be programmed to open and close its fingers, and thereby manipulate an object or carry out an assembly. Although a physical prototype of the gripper existed previously, the fact that our electronic version is simply a text file in a computer means that we can edit the file and easily change parameters such as finger number, length, and location. Figure 2, for example, illustrates a four-fingered version of the gripper. Thus, a designer

can experiment with many configurations before committing to a particular design.

Schematic editors. From our experience, conventional geometrical modelers are not practical for simulating large assemblies. For example, the time to model an automobile engine to the level of detail needed to verify properties such as combustion would be prohibitive. Thus, we are exploring the use of schematic editors.

Schematic editors let a designer study assemblies at various levels of abstraction. In the case of an automobile engine, a cylinder is represented by an axis, a piston head by a dot (giving the location of its coordinate system), and the crankshaft by an axis, as are certain points on the crankshaft that are constrained to be a fixed distance from the center of the axis. At this level, one can carry out a kinematic simulation of the engine. By adding mass and inertia, one can carry out a dynamic simulation. The cylinder and piston can be expanded into a geometric model to carry out a lower level simulation. If we were

interested in studying combustion properties, we could then further refine the model to put a finite element mesh in the cylinder and, using necessary engineering codes, study combustion properties.

Representing various parts of the overall system at different levels of abstraction allows very sophisticated simulations. However, much work remains to understand how models should be connected at various levels and how editing of one abstraction of the model is captured in another.

Model-driven simulator. A key component of the project is a model-driven simulator,² capable of supporting a wide range of multidisciplinary research, whose physical coverage can be extended and modified. The world of this system consists of a collection of rigid objects that are hinged together, move, and make contact with one another. A hinge is simply a constraint between a point in one coordinate system and a plane in another coordinate system. More complex hinges can be made out of these primitive hinges. For example, by

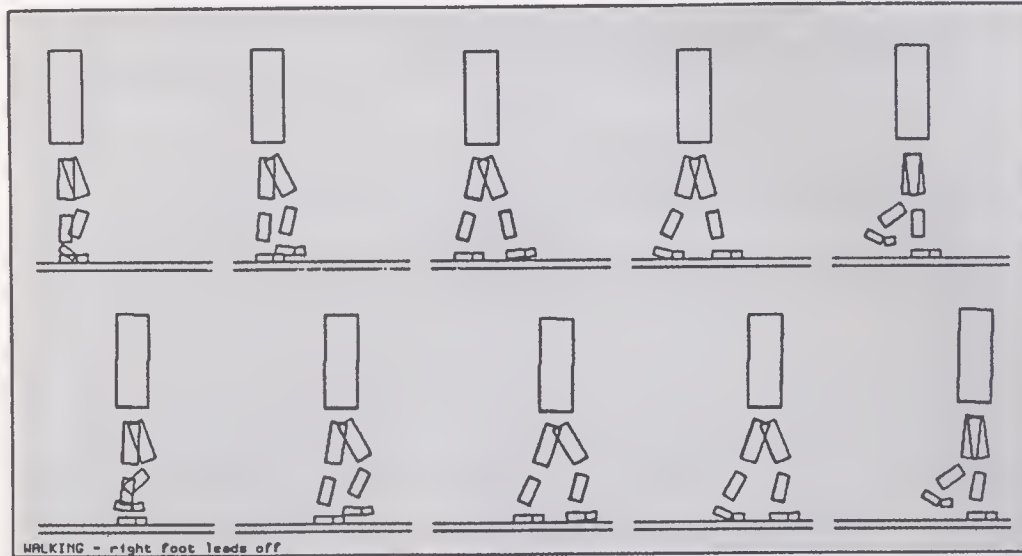


Figure 3. Walking figure is an example of the simulator's use to study high-degree-of-freedom control structures.

using three hinges, one can constrain a point in one coordinate system to be coincident with a point in another coordinate system.

Hinges come in two varieties: permanent hinges, which link two objects permanently together, and temporary hinges, which are inserted when two objects collide. Temporary hinges are necessary to modify the dynamic equations whenever a collision occurs. The system understands Newton's laws and, when a physical object is described, assembles equations such as force equals mass times acceleration. When an object collides with another, its motion is affected by the additional constraint that one object cannot penetrate another. Therefore, this additional constraint is added during a collision, and the system automatically regenerates the correct physical equations to describe the motion.

Perhaps the most important feature of our system is this ability to produce elaborate models of contact and collision. Such models are essential when carrying out simulations of many electronic proto-

types, such as our gripper in Figure 1, where friction and finger contact are critically important in manipulation algorithms.

The simulator has also been used to study control structures for high-degree-of-freedom objects. While programming a person to walk, shown in Figure 3, it was necessary to control balance and posture so that the frictional force of the foot would not cause the person to fall backward as the arm swung forward.

References

1. C. Hoffmann, J.H. Hopcroft, and M. Karasick, "Robust Set Operations on Polyhedral Solids," Tech. Report #87-875, Cornell Univ., Ithaca, New York, 1987.
2. C. Hoffmann and J.H. Hopcroft, "Simu-

Electronic laboratory. A long-range goal is the development of an electronic laboratory. The cost of a modest robotics laboratory is approximately \$300,000, but an electronic laboratory could be provided at a fraction of this cost. Electronic laboratories at many institutions could be coordinated with several institutions that housed the physical equipment. Students could debug their robotics programs electronically, and then verify their results by remote video on actual equipment. □

lation of Physical Systems from Geometric Models," *IEEE J. Robotics and Automation*, special issue on robot motion planning, Vol. RA-3, No. 3, June 1987, pp. 194-206.

Robotics Research at USC

George A. Bekey

*Computer Science Department, Henry Salvatori Bldg., Rm. 200,
University of Southern California, University Park, Los Angeles, CA 90089-0782*

The University of Southern California has been active in robotics for nearly a decade and has research projects involving faculty and graduate students in four departments.

Electrical engineering faculty members have worked in projects involving robot vision, teleoperator control, multifingered end effectors, robot skill acquisition, and sensor development. There are continuing major programs in image understanding and interpretation, manufacturing assembly, and telerobotics.

Mechanical engineering faculty members have developed a direct-drive manipulator and have worked on control of flexible robot structures and parallel computer implementation of control algorithms.

Industrial and systems engineering faculty members have developed model parts storage and retrieval systems and low-cost two- and three-degree-of-freedom manipulators. They have used simulation to analyze the effectiveness of robots in flexible manufacturing systems and have studied the economic aspects of robot integration. Expert systems in manufacturing, CAD/CAM, and fundamental aspects of engineering design are also being investigated.

Computer science faculty members have been active in studies involving geometric modeling for production automation, robot languages, application of neural networks to the computation of inverse kinematics and dynamics, as well as representation of material handling systems. They've also studied multiple intelligent robots in manufacturing and researched grasping control in both human and robot hands, including development of sensors, algorithms, and languages.

The research is funded by grants and contracts from the National Science Foundation, National Aeronautics and Space Administration, National Institutes of Health, Department of Defense Advanced Research Projects Agency, and industry. Industrial support comes primarily through the Institute for Manufacturing and Automation Research, a university-industry-government consortium. IMAR provides major funding for manufacturing-related research projects at USC and the University of California at Los Angeles, as well as some support to four other affiliated universities.

The research is carried on in various laboratories in the School of Engineering at USC, including the Robotics Research Laboratory, the Brain Simulation Laboratory, the Computer Vision Laboratory, and the Hand Research Laboratory. Facilities include a Puma 560 industrial robot, two IBM 7545 robots, a US Robotics 100 robot, two robots constructed by mechanical engineering faculty and students, several robot hands, a Watsmart motion tracking system, an excellent computer vision laboratory, and a number of high-performance workstations (Sun, Texas Instruments, and Symbolics).

The following sections describe some of the above projects and indicate the breadth of these activities. For further information, interested readers are urged to contact the individual faculty members mentioned.

Skill transfer from human to robot. Human beings perform highly skilled tasks, both in everyday life and in industrial settings. The goal of the skill-transfer project, directed by Sukhan Lee of the EE

Department, was to develop a system to transfer one such skill (peg insertion) from a human to a robot. A block diagram of the simulation used for this study is shown in Figure 1.

The system, known as EARS (expert-assisted robot skill acquisition), contains two major functional components concerned with skill transfer and learning, respectively. The transfer portion of EARS starts with the description of a situation in terms of an expert's language for describing the task and the robot's own internal language. The robot interprets the mismatch between its own skill and that of the expert and initiates a discovery process. The learning portion of the system uses rules as system background knowledge, constructs new rules from experience, and exploits self-exploration. The program, written in Lisp, uses the conceptual clustering approach to machine learning.^{1,2}

Telerobotics. One of the major problems in the design of telerobotic systems concerns the control of dual and redundant manipulators. Lee has developed a series of quality measures to determine optimal joint configurations and force distributions, optimal load balancing and operational stability, and robustness. These notions have been incorporated in new algorithms for dual arm control.

Manipulability is a measure of a robot's range, velocity, and acceleration capability in Cartesian space. Lee has extended this concept to the study of task-oriented, static and dynamic dual-arm systems.

In the area of teleoperation, work is proceeding on the following problems:

- control of teleoperators including the

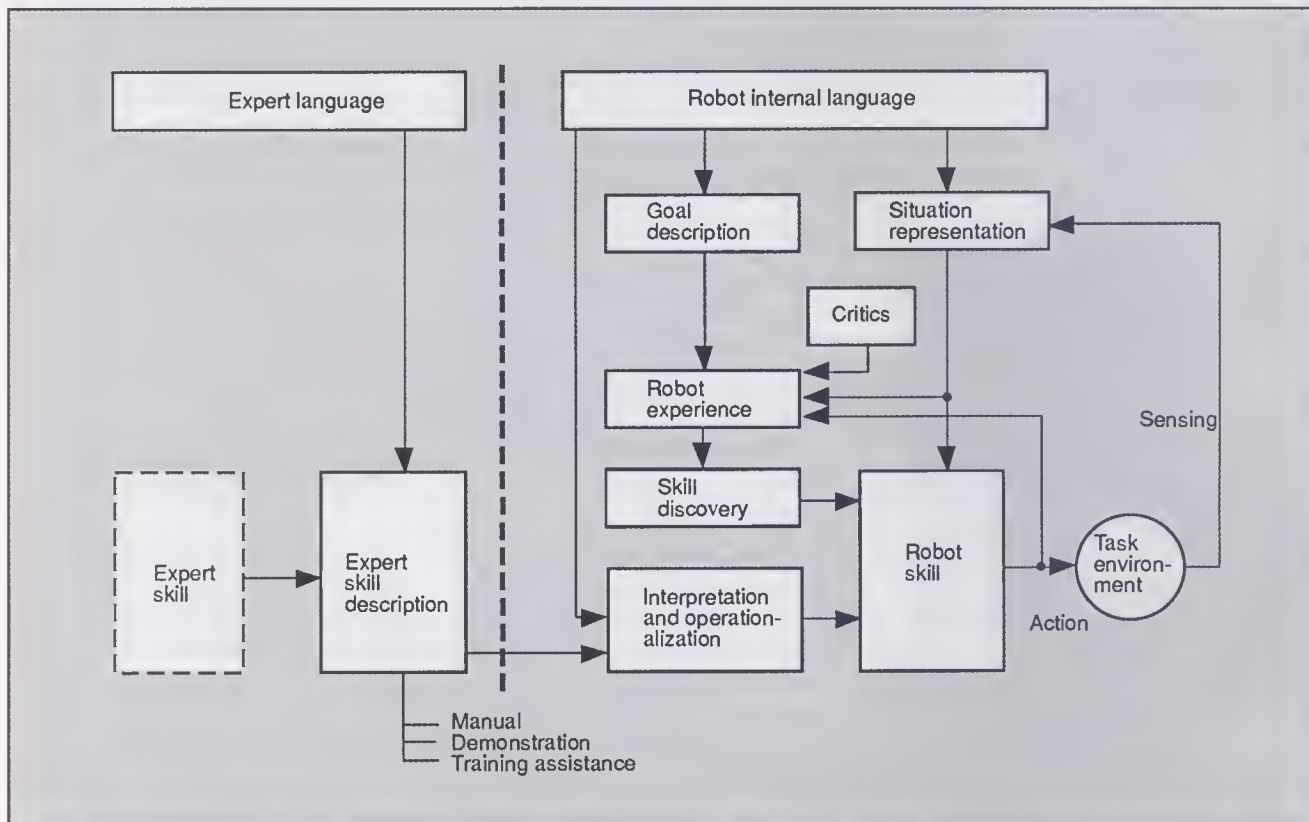


Figure 1. Model of expert-assisted robot skill acquisition.

effects of time delay in the transmission link,

- development of an expert system to assist the operator in the control of remote, redundant dual-arm teleoperators, and
- application of qualitative physics ideas from AI to develop symbolic models of teleoperation.

Work in this general area has been supported by NASA/JPL for several years.^{3,4}

Knowledge-based control of robot grasping. An alternative to solving finger-motion equations during a grasping task is to develop a knowledge base that includes the typical grasp poses of which humans are capable, along with their relation to the geometry of the target objects. Michael Arbib, Thea Iberall, and George A. Bekey of the CS Department have been

active on research problems of this nature for several years. In collaboration with Rajko Tomovic, a colleague in Yugoslavia, they have developed a nonnumerically controlled, five-fingered robot hand, known as the Belgrade-USC hand. A prototype of this hand is illustrated in Figure 2.

They are also involved in other research issues in connection with both human and robot hands, such as

- development of an appropriate language for grasp description, which is consistent with our knowledge of the brain structures involved in hand preshaping, trajectory control, and grasping;
- relation of task variables to grasp selection; and
- application of neural networks to the representation of grasp learning.^{5,6}

Hand-eye coordination. Gerard Medioni of the EE and CS departments is active in a number of research projects concerning computer vision and its applications. Specifically, he and his students have developed a range sensor-analysis system capable of yielding the generalized cone description of arbitrary solid objects. This description has been used as an input to Iberall and Bekey's grasp-selection system to develop a complete hand-eye coordination system. Figure 3 shows a block diagram of the system.⁷

Robot sensors. A number of research projects concern the development and evaluation of new sensors for robotic applications. Lee has developed a new proximity sensor using a hexagonal array of emitters and receivers, known as the Hexeye. A patent on this device has been applied for. The sensor will be used for

object localization and recognition.

Bekey and Ilija Zeljkovic, a research engineer in the USC Robotics Research Laboratory, are developing touch and slip-page sensors for the Belgrade-USC hand.



Figure 2. Prototype of the Belgrade-USC hand.

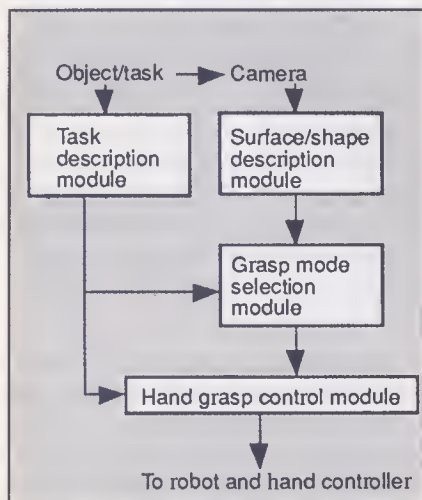


Figure 3. Block diagram of the robot hand-eye coordination system.

Prototypes of both sensors have been fabricated.

Control of flexible robot arms. Henryk Flashner of the Mechanical Engineering Department is concerned with modeling and control of flexible manipulators. His work has demonstrated the limits of performance for noncollocated control systems. He has obtained a theoretical basis for the stability of a large class of robot controllers (which include the traditional proportional-integral-derivative, or PID, controllers) and analyzed their robustness. He and his colleagues, Ali Pak and Geoffrey Shiflett, have also developed a direct-drive robot arm and studied its performance.^{8,9}

Low-cost robots for pick-and-place operations. Many industrial operations do not require the sophistication of a six-degree-of-freedom robot manipulator. Low-cost, highly reliable, simple robots

are preferable for such tasks, especially for high-volume repetitive applications. Behrokh Khoshnevis of the Industrial and Systems Engineering Department has developed and tested a family of such low-cost, simple, three-degree-of-freedom robotic devices.

Simulation of material handling components in manufacturing cells. Simulation languages such as Siman and Slam have nodes for representation of such processes as conveyor belts and automated guided vehicles. However, they do not have facilities for representation of robot manipulators. Khoshnevis and his students are developing high-level environments for simulation languages, where one single node can represent a robot. This node, along with parametric information on work envelopes, task, speed, and so on, can then be placed automatically into a simulation network.¹⁰ □

References

1. S. Lee and M.H. Kim, "Expert Assisted Robot Fine Motion Skill Acquisition," *Proc. Int'l Conf. AI, Expert Systems and Languages (IMCS)*, Barcelona, Spain, 1987, pp. 13-20.
2. S. Lee and M.H. Kim, "Learning Expert System for Robot Fine Motion Control," *Proc. 1988 IEEE Conf. Intelligent Control*, Arlington, Va., 1988 (in press).
3. S. Lee, G.A. Bekey, and A.K. Bejczy, "Computer Control of Space-Borne Teleoperators with Sensory Feedback," *Proc. 1985 IEEE Int'l Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., order number 615, pp. 205-214.
4. S. Lee, "Dual Redundant Arm Configuration Optimization with Task-Oriented Dual Arm Manipulability," *IEEE Trans. Robotics and Automation*, Vol. 5, Feb. 1989, pp. 78-97.
5. R. Tomovic, G.A. Bekey, and W.J. Karplus, "A Strategy for Grasp Synthesis with Multifingered Robot Hands," *Proc. 1987 IEEE Int'l Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., 3-vol. set, order number 787, pp. 83-89.
6. H. Liu, T. Iberall, and G.A. Bekey, "Building a Generic Architecture for Robot Hand Control," *Proc. IEEE Int'l Conf. Neural Networks*, San Diego, Calif., 1988, Vol. II, pp. 567-574.
7. K. Rao et al., "Robot Hand-Eye Coordination: Shape Description and Grasping," *Proc. 1988 IEEE Int'l Conf. Robotics and Automation*, Computer Society Press, Los Alamitos, Calif., 3-vol. set, order number 852, pp. 407-411.
8. V.A. Spector and H. Flashner, "Flexible Modeling for Control System Design," *Proc. AIAA Guidance, Navigation, and Control Conference*, Amer. Inst. of Aeronautics and Astronautics, New York, 1987, pp. 213-220.
9. V.A. Spector and H. Flashner, "Modeling of Non-Collocated Structural Control Systems," *Proc. AIAA Guidance, Navigation, and Control Conf.*, Amer. Inst. of Aeronautics and Astronautics, New York, 1988, pp. 74-83.
10. B. Khoshnevis, "An Expert Simulation Model Builder," *Proc. SCS Multi Conf. Intelligent Simulation Environment*, Soc. for Computer Simulation, San Diego, Calif., 1986, pp. 120-132.

Our Most Highly Developed Resource.

All of the technology at Northrop Defense Systems Division is the result of the intricacies of the human mind. The capacity to create is enormous. Its irrepressible drive to reach higher elements of knowledge is the direct source of Northrop's achievement. With these innovative ideas, Northrop maintains its leadership in advancing technology.

Software Engineers

Emphasis is on programming in the "C" and Ada languages following a structured design methodology and supported by graphics-based interactive development tools. Our state-of-the-art software development environment is implemented on a VAX cluster configuration, running under VMS, connected to SUN workstations on an Ethernet LAN, running under UNIX. Each software engineer has a terminal or workstation providing access to all computing resources on the network.

Our expanding software development requirements have created the following employment opportunities:

SOFTWARE DEVELOPMENT PROJECT MANAGERS

To lead design teams in the development of real-time operational flight programs and their subsequent integrating with the countermeasures hardware. Project responsibilities include the planning and management of schedules, budgets, and staffing requirements. Requires a Bachelor's degree in a technical field with computer science content and a minimum of eight years relevant experience, including experience in the development of large-scale "C" or Ada programs in a disciplined environment containing configuration management and quality assurance components. Advanced degree, formal management training, and avionics industry experience desirable.

SOFTWARE ARCHITECTS

To initiate design projects and perform requirements analyses, algorithm development, and high-level design. Requires an advanced technical degree with formal computer science training and five years experience in the development of large-scale real-time embedded software. Experience in structured design methodology in a military standards environment and exposure to knowledge-based expert-systems and object-oriented programming techniques desired.

SOFTWARE SYSTEMS DEVELOPMENT ENGINEERS

To participate as individual contributors or group leaders in the detailed design, coding, testing, and integration of embedded real-time software. Requires a BSCS, or equivalent, and three years experience in developing software in "C" or Ada using a structured design methodology. Knowledge of computer hardware architectures, performance simulation and modeling, and Kalman filtering algorithms is desirable.

Automatic Test Systems Engineers

We have openings for senior level engineering professionals to develop state-of-the-art automatic test stations for intermediate and depot level trouble-shooting and repair of military avionic systems.

MODULAR AUTOMATIC TEST EQUIPMENT (MATE) SYSTEMS

Tailoring/integrating MATE systems for control and support software. Includes MATE operating system/atlas compiler/test executive/on-line editor. Tailoring involves 1750 A assembly, JOVIAL, and Ada programming.

SOFTWARE ENGINEERING

Develop interactive graphic systems using object-oriented program data bases over distributed networks on high performance workstations.

ATS SYSTEMS ENGINEERING

Analysis, design (hardware/software), integration, support. Systems incorporate distributed networks with remote data base management.

TEST PROGRAM SET DEVELOPMENT

TPS design using automated test program generators, simulators and emulators for various high level software languages.

HARDWARE DESIGN

Analog, RF, microwave and digital circuit design for automated instrumentation including RF source and analog power supplies.

Please submit resume to: **Employment Office, Dept. C45, NORTHROP CORPORATION, 600 Hicks Road, Rolling Meadows, IL 60008.** We are an equal opportunity employer M/F/V/H. U.S. citizenship required for certain positions.

NORTHROP

Defense Systems Division
Electronics Systems Group



First Announcement and Call for Papers

First Conference on Visualization in Biomedical Computing

Atlanta, Georgia - May 22-25, 1990

Theme:

Visualization in scientific and engineering research is a rapidly emerging discipline aimed at developing tools and approaches to facilitate the interpretation of, and interaction with, large amounts of data, thereby allowing researchers to "see" and comprehend, in a new and deeper manner, the systems they are studying. This conference, the first of its kind, is aimed at bringing together researchers working in various aspects of visualization in order to present and discuss approaches, tools, and techniques associated with visualization science in general, and visualization in biomedical computing in particular.

Topics:

- Applications, techniques, tools
 - Visualization in
 - * Modeling and simulation of biomedical processes
 - * Diagnostic radiology
 - * Molecular biology and genetics
 - * Neurophysiology
 - * Prosthetics development
 - * Radiation treatment planning
 - * Education
- Theories and approaches
 - Models of visualization
 - Representation of multidimensional parameters
 - Psychophysical aspects of visualization
 - * Perception
 - * Cognition
 - * Human/machine interface
 - Artificial intelligence in visualization
 - Computer vision and image processing
 - Graphics and display
 - Visual communications and televisualization
 - Courses and training in visualization
- Other related topics

Sponsored by/in cooperation with:

- National Science Foundation
- Georgia Institute of Technology
- Emory University School of Medicine
- Emory-Georgia Tech Biomedical Technology Research Center

Organizing Committee:

- Dr. Norberto Ezquerra, Office of Interdisciplinary Programs, Georgia Institute of Technology (Chair).
- Dr. Ernest Garcia, Department of Radiology, Emory University.
- Prof. Ronald C. Arkin, School of Information and Computer Science, Georgia Institute of Technology.

Call for panels and tutorials:

Proposals for panels and tutorial courses are welcomed.

Deadlines:

- Paper Abstracts due: July 31, 1989 (Submit 4 copies of 800 word abstract)
- Panel and tutorial proposals due: July 31, 1989
- Acceptance notification to authors: November 1, 1989
- Full papers due: February 1, 1990
- Conference: May 22-25, 1990

For further information contact:

Dr. Norberto Ezquerra, Office of Interdisciplinary Programs, Georgia Tech, Atlanta, GA 30332. Phone: (404)-894-3964.

Secure Query-Processing Strategies

T.F. Keefe, University of Minnesota

M.B. Thuraisingham, Honeywell

W.T. Tsai, University of Minnesota

A multilevel secure database management system is a system that is secure when shared by users from more than one clearance level and contains data of more than one sensitivity level. MLS/DBMSs evolved from multilevel secure computing systems. Present-day DBMSs are not built with adequate controls and mechanisms to enforce a multilevel security policy. Thus, an MLS/DBMS is different from a conventional DBMS in at least the following ways:

(1) Every data item controlled by an MLS/DBMS is classified in one of several sensitivity levels that may need to change with time.

(2) Access to data must be controlled on the basis of each user's authorization to data at each sensitivity level.

Providing an MLS/DBMS on current secure computing systems presents a new set of problems. The most obvious is that the granularity of classification in a DBMS is generally finer than a file and may be as fine as a single data element. Another problem unique to databases is the necessity to classify data according to context, time, and aggregation. Furthermore, DBMSs are vulnerable to inference attacks—that is, users can infer unauthorized data from the knowledge they have accumulated.

**These techniques
allow users with
different clearance
levels to access data at
various sensitivity
levels without
violating the security
of a database
management system.**

The Air Force Summer Study of 1982¹ proposed various designs for multilevel secure relational database management systems. One MLS/RDBMS proposal is a near-term set of requirements incorporating off-the-shelf concepts; another is a longer-term set of requirements, including content, context, dynamic classifications, and solutions to the inference problem.

There has been a lot of interest in producing designs for MLS/RDBMSs.²⁻⁴ Much attention is also being paid to the foundations of the inference problem in database security.⁵⁻⁷ Other work has dealt with particular aspects of the inference problem,⁸ such as architectures for MLS/DBMSs and query-processing strategies.

The strategies for secure query processing that we propose here are carried out by query modification, a technique that has been used for enforcing integrity constraints and providing view mechanisms. The technique consists of replacing the query the user presents with one that, when evaluated, will perform the desired function. In the case of a view mechanism, the names of views referenced in the query are replaced by the definitions of the views in terms of base relations.

Security models for MLS/DBMSs are still a research issue.² Once an appropriate security model is chosen, a DBMS that adheres to it must be built. The correspondence between the model and the DBMS is demonstrated by verification. To allow an effective verification of the security properties of the DBMS, the security-relevant portion of the code must be small. Query modification allows the security function to be separated from the normal functions of a DBMS. It allows one to concentrate on the security properties of the

query modification algorithm. The security of the entire system can then be based on the verified security properties of the query modification algorithm and the correct, nonhostile behavior of the DBMS. This fact makes query modification a desirable approach for retrofitting security into an existing DBMS. However, demonstrating that the behavior of the DBMS is correct and nonhostile is expensive. In this article we do not consider this verification issue. Instead we concentrate on various strategies that can be used to modify the query.

A previous approach to secure query processing

Query modification techniques have been used extensively in the past to transform queries into a more appropriate format for a particular application. In deductive databases, queries against virtual relations, or relations that are not stored in the database but must be computed from rules in the rule base, have been transformed into queries against base relations.⁹ Query modification has been used in a discretionary access control mechanism.¹⁰ A variation of this technique was proposed for mandatory access control.⁴

In this section we will describe this query modification mechanism and some of its limitations. In essence, the user's query is modified by applying the relevant security constraints for the particular query in such a way that if the modified query is posed, then the response generated by the DBMS will not result in any security violation. An alternate approach to secure query processing would be not to modify the query but to check whether the response to be released will violate security. If it will, the response is not released. For this approach, methods of database integrity maintenance such as Sadri and Kowalski's¹¹ could be adapted.

Before illustrating the query modification technique with examples, we must first describe security constraints. Security constraints assign classification levels to all data in the database.⁴ They provide a basis for a versatile, powerful classification policy because any subset of data can be specified and assigned a level statically or dynamically. Simple constraints provide for classification of the entire database, as well as classification by relation and by attribute. Content-based constraints provide a classification based on

the contents of the database and allow for classification by tuple and by element. Context-based constraints classify relationships between data. Finally, the classification levels of the data can change dynamically with changes in content or context.

A constraint consists of a data specification and a classification. The data specification defines a subset of the database and can be expressed with relational algebra. The classification defines the classification level of this subset. For example, consider a database that consists of a relation Emp(SS#, Name, Salary, Dept#) with SS# as the key. The content-based constraint that classifies the names of all employees who earn more than 100K (\$100,000 a year) as Secret is expressed

Level(PROJECT[Name](SELECT [Salary > 100K] Emp)) = Secret

and the context-based constraint that classifies all names and salaries taken together as Secret is expressed

Level(PROJECT[Name, Salary] Emp) = Secret

The simple constraints that classify all names and salaries in Emp taken individually as Unclassified are expressed

Level(PROJECT[Name] Emp) = Unclassified

Level(PROJECT[Salary] Emp) = Unclassified

Simple and content-based constraints classify elements of tuples in the database and therefore can be applied to data as it is actually stored in the database. In this case, each element of data is associated with its sensitivity level. Context-based constraints classify sets of elements in the database and can be applied only in the computation of the result that is to be output in response to a user's query. For example, if the content-based constraint that classifies all names of employees who earn more than 100K as Secret is enforced, the Secret names are stored in a Secret file and the Unclassified names are stored in an Unclassified file. If the context-based constraint that names and salaries taken together are Secret is enforced, the names and salaries are stored in Secret files and the constraint allows the information to be downgraded in appropriate situations.

The security constraints are used in evaluating queries as follows: Suppose an unclassified user wants to retrieve all the names from the relation Emp, and the content-based constraint that classifies all names of employees who earn more than 100K is enforced. Then the query is modi-

fied to retrieve all names of which the corresponding salaries are less than or equal to 100K. This modified query is then evaluated and all the Unclassified names will be retrieved. The formal description of this query modification algorithm appears in an article by Dwyer, Jelatis, and Thuraisingham.⁴

Although the query modification algorithm modifies the query correctly, it does not specify how to extract the relevant constraints. In some cases, a relation may have numerous constraints enforced on it. In addition, constraints enforced on Join operations are rather complex. For example, consider the following relations:

Emp(SS#, Name, Salary, Dept#)

Dept(Dept#, Project, DName, Mgr)

SS# is the key of Emp, and Dept# is the key of Dept. We enforce the constraint that classifies all names in Emp who work in a particular project—say Stars—in Dept as Secret. This constraint necessitates the Join of the relations Emp and Dept. Such constraints necessitate an efficient search procedure to extract the relevant constraints for a query.

Another drawback of the query modification algorithm is that it takes into consideration only the response to the current query. It does not maintain the history information, making it impossible to determine whether the response to be generated, taken together with previously released responses, might lead to security violations. In other words, the algorithm does not handle inference attacks. To overcome such attacks, the information in all responses released must be maintained. Furthermore, for each additional query some efficient computation technique is required to see if the response generated, combined with previous responses, will cause a breach in security.

In the next section we will explore the representation of constraints and the representation of histories of released information, along with methods for manipulating them to process queries securely.

Strategies for secure query processing

There are two important concerns about secure query-processing strategies. The first is the representation of the information required for processing the query. This information consists of security constraints, including simple, content-based, and context-based types, and environmen-

tal state information concerning what data has been released and at what level. The second concern is efficient methods of manipulating this information—for example, methods for finding constraints relevant to a given query and methods for constructing the modified query.

Basic strategy. Most existing relational database management systems use the relational model to represent metadata. Therefore, we will first examine the suitability of such a model for the representation of metadata for an MLS/DBMS. The metadata of an MLS/DBMS includes the external schemas that describe the views, the conceptual schemas that describe the relations and the attributes, the internal schemas that describe the physical files, the security constraints, and the integrity constraints.

We will illustrate our design of the data dictionary, which contains the metadata, with an example. Suppose the database consists of the relations Emp and Dept described earlier. Let the following security constraints be enforced:

- C1: Name in Emp where Salary > 100K is Secret.
- C2: Name and Salary taken together are Secret.
- C3: Name in Emp where Project in Dept is Stars is Secret.
- C4: DName and Mgr taken together when Dept# = D1 is Secret.
- C5: Project in Dept is Secret.

These constraints can be expressed as discussed in the preceding section. Their representation is as follows:

- C1: Level(PROJECT[Name](SELECT [Salary > 100K] Emp)) = Secret
- C2: Level(PROJECT[Name, Salary]Emp) = Secret
- C3: Level(PROJECT[Name](SELECT [Project = Stars](Emp JOIN Dept))) = Secret
- C4: Level(PROJECT[DName, Mgr](SELECT [Dept# = D1] Dept)) = Secret
- C5: Level(PROJECT[Project] Dept) = Secret

The representation of this metadata is shown in Tables 1 through 5. Table 1 describes the relations, the arity of each relation, the number of tuples in each relation, one primary key, and one foreign key. Table 2 describes the attributes in each relation. Table 3 describes the views. Table 4 describes the content-based and simple constraints. Note that a simple constraint is a content-based constraint with no condition. Table 5 describes the context-based constraints. We have not

Table 1. Relation definitions.

Rname	Arity	Cardinality	Primary Key	Foreign Key
Emp	4	0	SS#	Dept#
Dept	4	0	Dept#	Null

described the internal schemas as they are not required for the query modification process.

The proposed strategy has two major weaknesses. The first is that it doesn't maintain any environmental state information. To examine this weakness, let's suppose an unclassified user wants to obtain all names. Tables 4 and 5 will have to be searched for all the relevant constraints. The relevant constraints in this case are C1, C2, and C3. Furthermore, C1 and C3 are directly relevant, and the query first will be modified to retrieve all names of employees whose salary is less than 100K and who do not work in the project Stars. The constraint C2 becomes directly relevant only if the salaries have been released earlier. But this information is not represented in the model.

The second weakness of this strategy is its difficulty in expressing complex secu-

Table 2. Attribute definitions.

Rname	Aname
Emp	SS#
Emp	Name
Emp	Salary
Emp	Dept#
Dept	Dept#
Dept	Project
Dept	Dname
Dept	Mgr

rity classifications. We encounter this difficulty in representing constraints in which the sensitivity level of one attribute depends on the sensitivity level of another. In the previous example, suppose we have another constraint, which classifies salary as Secret only if Dept# in Emp is Secret. If an unclassified user wants to retrieve the

Table 3. View definitions.

View Name	View Def
V1	PROJECT [SS#, Name, Salary, Dept#, Project, Dname, Mgr] (Emp JOIN Dept)
V2	PROJECT [Name, Salary] (Emp)
V3	PROJECT [Dname, Mgr] (Dept)

Table 4. Simple and content-based constraints.

Rname	Cname	Condition	Target	Level
Emp	C1	Salary > 100K	Name	Secret
V1	C3	Project = Stars	Name	Secret
Dept	C5	Null	Project	Secret

Table 5. Context-based constraints.

Rname	Cname	Condition	Target	Level
Emp	C2	Null	Name	Secret
Emp	C2	Null	Salary	Secret
Dept	C4	Dept# = D1	Dname	Secret
Dept	C4	Dept# = D1	Mgr	Secret

salaries, then the query is first modified to retrieve all salaries for which the predicate Dept#_Secret is False. But this modified query cannot yet be evaluated against the relational database. The tables have to be searched again to get all the constraints that classify the department number as Secret to provide a definition for the Dept#_Secret predicate.

Strategy 2: adding environmental information. Here we will build on the basic strategy and add the environmental information it lacked. We will express this strategy using logic. Logic can be used in two ways: to express a computation and to imply a method of execution. We use logic in both ways. First, we express the computation performed by the strategy. Second, we analyze the performance of the strategy, assuming the implied method of execution. We assume a method of execution based on a technique proposed by Chang for translating queries against virtual relations to queries against base relations.⁹

We will now describe how the constraints defined in the preceding subsection can be expressed in logic. A logical formula is of the form $A \rightarrow B$, meaning A implies B where A is called the premise and B the conclusion. Each base relation (a relation in the database) or virtual relation (a relation not stored in the database) is represented by a predicate. Therefore, the constraints C1 through C5 are expressed by the following rules:

- R1: $\text{Emp}(X,Y,Z,W) \wedge Z > 100K \rightarrow \text{Level}(Y, \text{Secret})$
- R2: $\text{Emp}(X,Y,Z,W) \rightarrow \text{Level}([Y,Z], \text{Secret})$
- R3: $\text{Emp}(X,Y,Z,W) \wedge \text{Dept}(P,Q,R,S) \wedge P = W \wedge Q = \text{Stars} \rightarrow \text{Level}(Y, \text{Secret})$
- R4: $\text{Dept}(P,Q,R,S) \wedge P = D1 \rightarrow \text{Level}([R,S], \text{Secret})$
- R5: $\text{Dept}(P,Q,R,S) \rightarrow \text{Level}(Q, \text{Secret})$

The rules R2 and R4 are context-based constraints. They classify sets of attributes released together. Therefore, if one of the attributes has already been released, then the other should be classified at the Secret level. We have found that with the relational model it was not easy to express such constraints. We will now examine how they can be expressed with logic. First we will examine the case where only one variable is released in a query. For this case the constraints R2 and R4 can be expressed as follows:

- R6: $\text{Emp}(X,Y,Z,W) \wedge \text{Release}(Y, \text{Unclassified}) \rightarrow \text{Level}(Z, \text{Secret})$

- R7: $\text{Emp}(X,Y,Z,W) \wedge \text{Release}(Z, \text{Unclassified}) \rightarrow \text{Level}(Y, \text{Secret})$
- R8: $\text{Dept}(P,Q,R,S) \wedge P = D1 \wedge \text{Release}(R, \text{Unclassified}) \rightarrow \text{Level}(S, \text{Secret})$
- R9: $\text{Dept}(P,Q,R,S) \wedge P = D1 \wedge \text{Release}(S, \text{Unclassified}) \rightarrow \text{Level}(R, \text{Secret})$

where $\text{Release}(X,L)$ implies X has been released at level L .

Suppose an unclassified user wants to retrieve all names from Emp:

- $\{Y: \text{Emp}(X,Y,Z,W)\}$

Instead of modifying the query to obtain all Unclassified names, we pose a query that will first obtain all names, then obtain Secret names, and finally subtract the Secret names from the set of all names. In general, this will give only a subset of the Unclassified names because Name is not a key attribute. The situation could arise in which two different employees share the same name and one makes less than 100K and the other makes more than 100K. According to R1, the name the two employees share is both Secret and Unclassified. However, the subtraction of the two sets would remove the name. The user will get only the Unclassified names.

More specifically, in processing the query, the system searches the rule base for all rules that classify the names in Emp with a sensitivity level higher than the clearance of the user. These rules are R1, R3, and R7. The premises in these rules are conjuncted and the result is used to form a query Q' . This query is as follows:

- $\{Y: (\text{Emp}(X,Y,Z,W) \wedge Z > 100K) \vee (\text{Emp}(X,Y,Z,W) \wedge \text{Dept}(P,Q,R,S) \wedge P = W \wedge Q = \text{Stars}) \vee (\text{Emp}(X,Y,Z,W) \wedge \text{Release}(Z, \text{Unclassified}))\}$

The query Q' is subtracted from the original to give the Unclassified names. The result is as follows:

- $\{Y: \text{Emp}(X,Y,Z,W)\} \text{ Minus } \{Y: (\text{Emp}(X,Y,Z,W) \wedge Z > 100K) \vee (\text{Emp}(X,Y,Z,W) \wedge \text{Dept}(P,Q,R,S) \wedge P = W \wedge Q = \text{Stars}) \vee (\text{Emp}(X,Y,Z,W) \wedge \text{Release}(Z, \text{Unclassified}))\}$

This query still cannot be evaluated because Release is not a base relation. Therefore, the rule base will be searched to see if the salaries have been released. Since the salaries have not yet been released, $\text{Release}(Z, \text{Unclassified})$ is False. Therefore, the query that will be evaluated is

- $\{Y: \text{Emp}(X,Y,Z,W)\} \text{ Minus } \{Y: (\text{Emp}(X,Y,Z,W) \wedge Z > 100K) \vee$

- $(\text{Emp}(X,Y,Z,W) \wedge \text{Dept}(P,Q,R,S) \wedge P = W \wedge Q = \text{Stars})\}$

This query will be translated into a language appropriate to the DBMS. It will be evaluated against the relational DBMS, and the response will be released. Since processing the query releases names at the Unclassified level, this information has to be recorded as environmental state information. We do this by adding the rule

- R10: $\text{Emp}(X,Y,Z,W) \rightarrow \text{Release}(Y, \text{Unclassified})$

to the rule base. This rule is not completely accurate; it implies that all the names were released at the Unclassified level. It does, however, err on the conservative side and is more efficient than other, more accurate ways of recording this information.

Now if the unclassified user wants to obtain the salaries in Emp, the query posed will be

- $\{Z: \text{Emp}(X,Y,Z,W)\}$

This query will be modified to

- $\{Z: \text{Emp}(X,Y,Z,W)\} \text{ Minus } \{Z: \text{Emp}(X,Y,Z,W) \wedge \text{Release}(Y, \text{Unclassified})\}$

However, by rule R10, $\text{Release}(Y, \text{Unclassified})$ is True and the query will release no data.

In the case where a query can release more than one attribute, a slightly more complicated method is necessary. This method deals with the problems of how a set of attributes is given a classification and how the sensitivity level of a set of attributes is computed. The constraints R2, R6, and R7 represent the more general constraint C2, which says that Name and Salary taken together are Secret. R2 classifies the set {Name, Salary} as Secret if they are to be released simultaneously. R6 classifies Salary as Secret if Name has previously been released. R7 classifies Name as Secret if Salary has already been released. In general, to classify a set of objects, it is necessary to have one classification describing each subset of the classified set. For example, consider the problem of classifying the set of objects {A, B, C} in relation X as Secret. The following classifications must exist:

- E1: $X(A,B,C) \rightarrow \text{Level}([A,B,C], \text{Secret})$
- E2: $X(A,B,C) \wedge \text{Release}(A, \text{Unclassified}) \rightarrow \text{Level}([B,C], \text{Secret})$
- E3: $X(A,B,C) \wedge \text{Release}(B, \text{Unclassified}) \rightarrow \text{Level}([A,C], \text{Secret})$
- E4: $X(A,B,C) \wedge \text{Release}(C, \text{Unclassified}) \rightarrow \text{Level}([A,B], \text{Secret})$
- E5: $X(A,B,C) \wedge \text{Release}(A, \text{Unclassified}) \wedge \text{Release}(B, \text{Unclassified}) \rightarrow \text{Level}(C, \text{Secret})$

E6: $X(A,B,C) \wedge \text{Release}(A, \text{Unclassified}) \wedge \text{Release}(C, \text{Unclassified}) \rightarrow \text{Level}(B, \text{Secret})$
 E7: $X(A,B,C) \wedge \text{Release}(B, \text{Unclassified}) \wedge \text{Release}(C, \text{Unclassified}) \rightarrow \text{Level}(A, \text{Secret})$

The number of rules is exponential with respect to the number of elements in the classified set. This is a significant weakness of this strategy if there are large sets of attributes involved in context-based security constraints.

The next problem we need to address is how to compute the sensitivity level of the list of attributes being released by a query. The answer is that the level for the set of attributes dominates the level assigned to each subset. For example, if the constraints E1 through E7 are in force, any set of attributes that contains the set $\{A,B,C\}$ will be classified as Secret. If C has been released Unclassified, then any set of attributes that includes $\{A,B\}$ is also Secret. Since each subset must be classified, the process takes exponential time with respect to the number of attributes released by the query.

The examples described above show that unlike the basic strategy, this one can be used fairly easily to represent constraints and environmental information. It allows a natural expression of context-based constraints and classification of elements based on the classification of other elements. However, the method is inefficient in representing context-based constraints and computing the classifications of sets of output attributes.

Strategy 3: using graphs. This strategy builds on the previous one in two ways. First, it uses graphs to index security constraints and make access more efficient. Second, it processes each output attribute independently, reducing the difficulty in processing queries with multiple output attributes and reducing the number of rules needed to express context-based constraints. This technique is a variation of the factoring mechanism used for query processing. The essence of the factoring mechanism is to extract the common information present in all the rules to avoid representation of redundant information.¹² For example, if more than one constraint is enforced on the relation Emp, then it is not necessary to repeat the name Emp and its attributes in each constraint. Just one point represents Emp, and the constraints enforced on it originate at this point.

Figure 1 shows how the constraint C1 is represented. Emp(X,Y,Z,W) is the label of a node with in-degree 0. Since Emp has four attributes, four edges originate at this node. The nodes at the end of these edges specify the corresponding attribute. For example, Name(Y) is the label of the node that states that Y in Emp is the attribute Name. The condition $Z > 100K$ in constraint C1 is represented by a node that points to Salary(Z). Since C1 doesn't assign levels to the attributes SS# and Dept#, no edges originate at the nodes labeled SS#(X) and Dept#(W). The constraint C1 classifies all names of employees who earn more than 100K as Secret. Therefore, there is an edge originating from Name(Y) whose tail is the node labeled Level(Y, Secret). Level(Y, Secret) points to the node with the condition $Z > 100K$, which classifies the name.

When an unclassified user wants to retrieve the names only, the search will be for a node that classifies names at a level greater than or incomparable to the Unclassified level. The node Level(Y, Secret) is such a node. Then all the nodes pointed to by edges emanating from Level(Y, Secret) will be examined and the conditions in those nodes will be negated

to modify the query. Therefore, the resultant query will ask to retrieve all names of employees whose salaries are less than or equal to 100K. If a Secret user wants to retrieve all the names, then the search will be for a node that classifies all names at a level greater than or incomparable to the Secret level. Since there is no such node, the query will not be modified.

Handling virtual relations. Initially we assumed that there are no constraints on derived relations such as the constraint C3, nor can we handle multiple queries such as retrieving the names first and then the salaries. We will now examine each of these cases. First, suppose the constraint C3 is enforced in addition to C1. The additional information that has to be incorporated into Figure 1 is shown in Figure 2. To express the constraint C3, first a join operation has to be performed between Emp and Dept over the attribute Dept#. Therefore, we have another node Dept(P,Q,R,S) that has in-degree 0 and four edges originating from it to represent its four attributes. Since the join is over Dept#, an edge from the node Dept#(W) that corresponds to the relation Emp, an edge from Dept#(P) that corresponds to

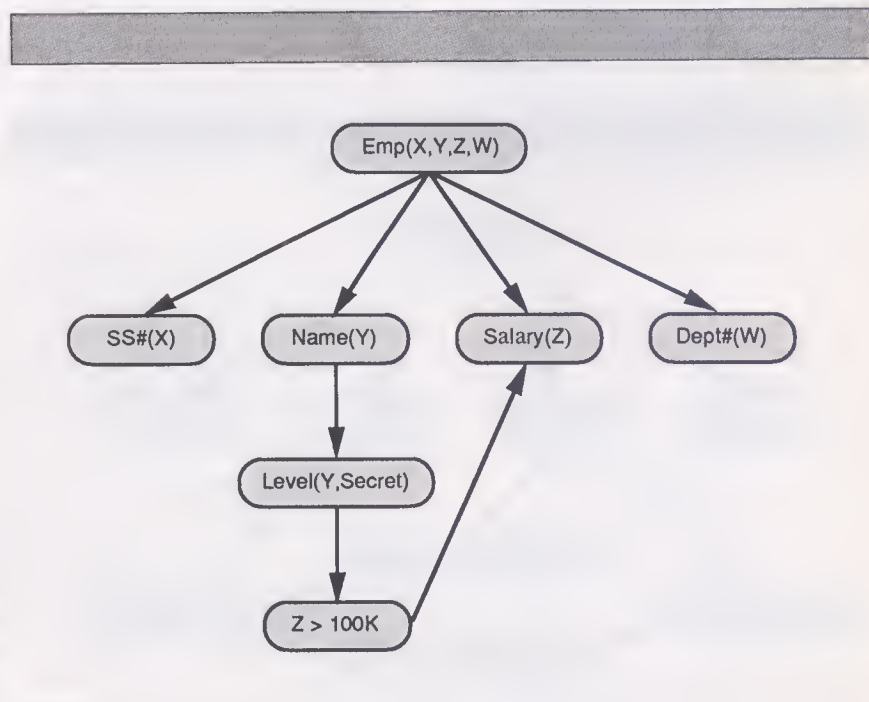


Figure 1. Representation of constraints with a graph.

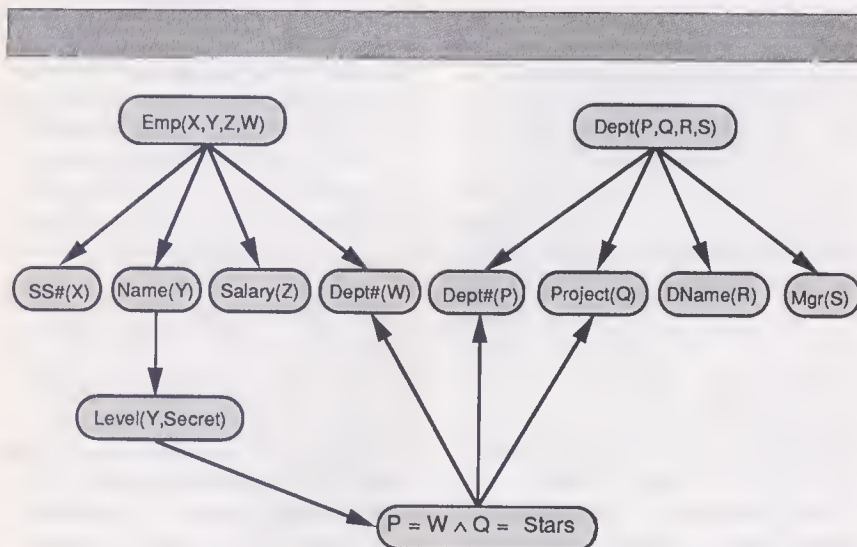


Figure 2. Constraints on joins.

relation Dept, and an edge from Project(Q) all meet at the node $P = W \wedge Q = \text{Stars}$. Since the names in project Stars are Secret, there is an edge from Level(Y, Secret) to the node $P = W \wedge Q = \text{Stars}$.

The query is processed in a manner similar to before. For example, if an unclassified

user wants to retrieve all names, a search is made for a node that classifies all names at a level greater than or incomparable to the Unclassified level. In this example the node found will be Level(Y, Secret). Then all edges originating from this node will be traversed. The two nodes

that will specify the conditions to classify the names are $Z > 100K$ and $P = W \wedge Q = \text{Stars}$. Then these conditions will be negated and the query will be modified to retrieve all names for whom the salary is less than or equal to 100K and who do not work in Stars.

Handling multiple queries. Now let us examine how multiple queries can be handled. Because of the constraint C2, names should not be released to an unclassified user if salaries have been released earlier, and vice versa. In other words, the context-based constraint C2 will automatically enforce the two constraints R8 and R9; names are Secret if salaries have been released, and salaries are Secret if names have been released. The additional information that has to be incorporated into Figure 1 to enforce the constraints R8 and R9 is shown in Figure 3. In this figure we have two more nodes. One is labeled Release(Y, Unclassified), which implies that names are released; the other node is labeled Release(Z, Unclassified), which implies that salaries are released. Because of the constraint R8, there will be an edge from Level(Y, Secret) to the condition Release(Z, Unclassified), and the constraint R9 will be enforced by the edge from Level(Z, Secret) to the condition Release(Y, Unclassified). Now if an unclassified user wants to obtain salaries after he has obtained names, a search will be made for a node that classifies salaries at a level greater than or incomparable to the Unclassified level. The node that will be found is Level(Z, Secret). The condition pointed to by this node is Release(Y, Unclassified). Therefore, the query will be modified to retrieve all salaries if names have not been released.

But this query still cannot be evaluated because the predicate Release is a virtual predicate. Therefore, a search must be made to determine whether names have been released. To do this, we need another graph to represent environmental information. This graph is shown in Figure 4. As soon as the names are released at the Unclassified level from the previous query, this information has to be incorporated into the environmental information. Therefore, the Name(Y) in Figure 3 has an edge whose tail is Release(Y). Since the names are already released at the Unclassified level, the negation of the condition Release(Y) is False. Therefore, the query is modified to the null query and no query will be posed to retrieve the salaries.

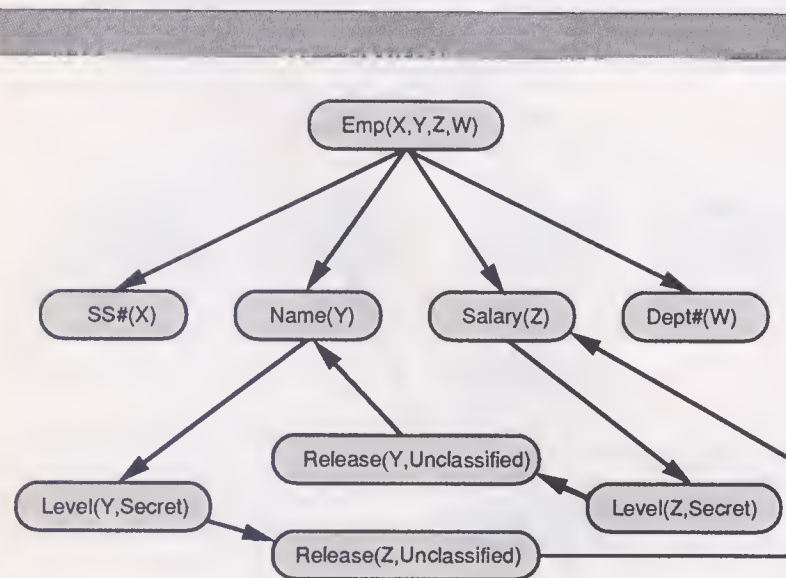


Figure 3. Representing context-based constraints.

Multiple outputs released in a query. Strategy 2 was inefficient in dealing with context-based constraints and the release of multiple outputs in a query. Up to this point we have only sketched the method for handling these problems by graph factoring. In this section we show how sets of outputs can be classified efficiently and how context-based constraints can be represented efficiently.

In this strategy the modification of the query necessary for each output variable is considered independently. As soon as the query is modified, the proper attribute is tentatively marked *released* in the environmental state graph. Each output is processed in this fashion. The modified query is processed. If the query is a contradiction, then no data is released and the changes to the environmental state graph are removed. If the query does release data to the user, the changes to the environmental state graph are made permanent. For example, suppose an unclassified user attempts to retrieve name and salary in a query. First, the release of names is considered. The condition for releasing names Unclassified is $\neg \text{Release}(\text{Salary}, \text{Unclassified})$. This predicate is immediately looked up in the environmental state graph and replaced with $\neg \text{False}$ or True . This condition is added to the query. At this point the fact that names have been released Unclassified is added tentatively to the environmental state graph. Next, the release of salaries is considered. The condition for releasing them is $\neg \text{Release}(\text{Name}, \text{Unclassified})$. This is looked up and replaced by $\neg \text{True}$ or False . Now this condition is added to the query, and the environmental state graph is again tentatively updated. This modified query is a contradiction. When it is processed, it will not release any data and so the environmental state graph will be returned to its previous state.

Since each output is processed individually, the classifications for context-based constraints are much simpler. For each object in a context-based constraint, there must be one node with a predicate that is the conjunction of release predicates for each of the other members of the constraint. This means that for a context-based constraint with n members, there will be a total of n predicate nodes in the tree with $n * (n - 1)$ logical terms in all of the predicates. This is much better than the previous strategy's exponential growth of classification rules with n .

The mechanism described here is also applicable when the level of a piece of data depends on the level of another. Suppose

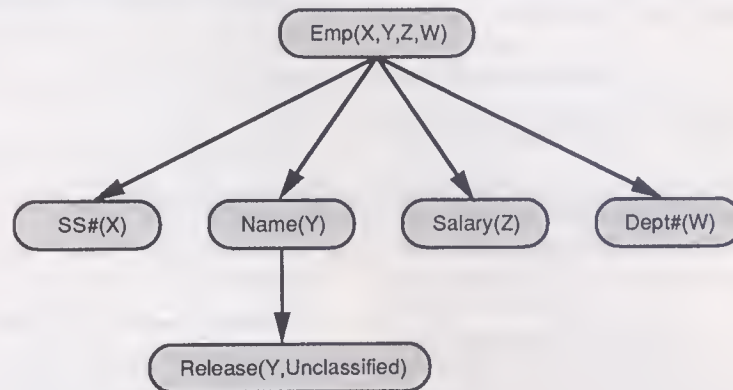


Figure 4. Representing environmental information.

the names in Emp are Secret only if the salaries are Secret. Then the node labeled $\text{Level}(\text{Y}, \text{Secret})$ has an arrow to the node labeled $\text{Level}(\text{Z}, \text{Secret})$. The query is then modified to retrieve all names for which the salary is not Secret. This query cannot yet be evaluated as it contains a virtual relation. The graph is searched again for all the conditions that make salary in Emp Secret. This is quite simple—all one has to do is follow all the edges that originate at $\text{Level}(\text{Z}, \text{Secret})$.

Performance

The performance of a query-processing strategy depends on the modification method being used, the number of security constraints, and the number of attributes released in the query, among other factors. That is the type of performance we focus on in this section. Another measure of performance is the impact of query modification on throughput. This type of performance is affected by such factors as the DBMS architecture (distributed or centralized), the type of queries being made, and the frequency of updates. It is not affected by the particular method used to modify the query. We will not consider that type of performance here.

Strategy 2. Security constraints in the second strategy are represented as rules. The method described by Chang⁹ for identifying relevant queries requires that

each literal in the query be unified with each security constraint. Performance is directly related to the number of security constraints. Thus, performance may be poor because there can be a large number of security constraints. Performance is degraded further due to the way that context-based constraints are represented. As discussed earlier, the number of security constraint rules necessary to represent a context-based constraint is exponential to the number of objects classified by the constraint.

The performance of constraint location is determined by the method of classification used for sets of output variables. The number of conditions added to the modified query is exponential to the number of variables being simultaneously queried. For the query

$\{A, B, C: (\text{Emp}(A, B, C, D))\}$

The augmented query Q' will be

$\{A, B, C: (\text{Emp}(A, B, C, D) \\ \wedge \text{Level}([A, B, C], \text{Unclassified}) \\ \wedge \text{Level}([A, B], \text{Unclassified}) \\ \wedge \text{Level}([A, C], \text{Unclassified}) \\ \wedge \text{Level}([B, C], \text{Unclassified}) \\ \wedge \text{Level}(A, \text{Unclassified}) \\ \wedge \text{Level}(B, \text{Unclassified}) \\ \wedge \text{Level}(C, \text{Unclassified}))\}$

and this query will determine the number of literals that will be resolved with the security constraints. In the best case, with no context-based constraints and only single outputs released simultaneously, the time complexity for location of constraints

is $O(n)$ where n is the number of security constraints.

Strategy 3. In the third strategy each classification constraint is associated with one attribute of a relation. In effect the constraints are indexed by the relation and the attribute. Since the attributes used in processing a query are evident from inspecting the query, this indexing makes the search for the relevant security constraints efficient. As pointed out earlier, the way the query modification is done allows each output to be handled independently. This means that graphs representing context-based constraints with groups of n members have space complexity of $O(n^2)$. The performance of the query modification process depends directly on the number of constraints that apply to the output attributes and not on the number of security constraints overall.

In this article we have described strategies for query processing in an MLS/DBMS: a basic strategy making use of query modification and two extensions of this technique, Strategy 2 and Strategy 3. Strategy 2 is based on logic and adds environmental state information to deal with context-based constraints. Strategy 3 uses graphs to index security constraints and processes output variables independently. It adds no new capabilities but provides increased performance. Both techniques ensure that a query is modified in such a way that if the modified query is posed, the response generated will not violate security requirements.

We have examined performance issues for the two proposed strategies. The indexing of security constraints in graph-based Strategy 3 makes it on the whole more efficient. However, logic-based Strategy 2 can support more flexible classifications of data. Strategy 3 is clearly superior in dealing with context-based constraints, generating fewer classification rules per constraint and processing queries with multiple outputs more efficiently.

There are many problems in the development of an MLS/DBMS. Among them are developing a security model, defining strategies for query processing, verifying the security properties of the system, and assuring adequate system performance. Secure query processing can form an essential part of an MLS/DBMS. □

References

1. "Multilevel Data Management Security," Committee on Multilevel Data Management Security, Air Force Studies Board, Commission on Engineering and Technical Systems, National Research Council, National Academy Press, Washington, DC, 1983.
2. D.E. Denning et al., "Secure Distributed Data Views: The Sea View Formal Security Policy Model," A003: Interim Report, SRI International, 1987.
3. B.B. Dillaway and J.T. Haigh, "A Practical Design for Multilevel Security in Secure Database Management Systems," *Proc. Aerospace Security Conf.*, 1986, pp. 44-57.
4. P.A. Dwyer, G.D. Jelatis, and M.B. Thuraishingham, "Multilevel Security in Database Management Systems," *Computers & Security J.*, Vol. 6, No. 3, June 1987, pp. 252-260.
5. M. Morgenstern, "Controlling Logical Inference in Multilevel Database Systems," *Proc. 1988 Symp. Security and Privacy*, Computer Society Press, Los Alamitos, Calif., order number 850, pp. 245-255.
6. T.H. Hinke, "Inference Aggregation Detection in Database Management Systems," *Proc. 1988 Symp. Security and Privacy*, Computer Society Press, Los Alamitos, Calif., order number 850, pp. 96-106.
7. T. Su and G. Ozsoyoglu, "Data Dependencies and Inference Control in Multilevel Relational Database Systems," *Proc. 1987 Symp. Security and Privacy*, Computer Society Press, Los Alamitos, Calif., order number 771, pp. 202-211.
8. M.B. Thuraishingham, "Security Checking in Relational Database Management Systems Augmented with Inference Engines," *Computers & Security J.*, Vol. 6, No. 6, Dec. 1987, pp. 479-492.
9. C.L. Chang, "Deduce 2: Further Investigations of Deduction in Relational Data Bases," in *Logic and Data Bases*, Herve Gallaire and Jack Minker, eds., Plenum Press, New York, 1978, pp. 201-236.
10. M. Stonebraker and E. Wong, "Access Control in a Relational Database Management System by Query Modification," *Proc. 1974 ACM National Conf.*, ACM, New York, pp. 180-187.
11. F. Sadri and R. Kowalski, "A Theorem-Proving Approach to Database Integrity" in *Foundations of Deductive Databases and Logic Programming*, Jack Minker, ed., Morgan Kaufmann Publishers, Los Altos, Calif., 1988, pp. 313-362.
12. W.A. Woods, "Knowledge Base Retrieval" in *Knowledge Base Management Systems*, M. Brodie and J. Mylopoulos, eds., Springer-Verlag, 1986, pp. 179-198.

Readers may write Thuraishingham at Mitre Corporation, K325, Burlington Road, Bedford, MA 01730.

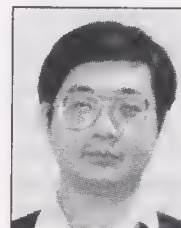


T. F. Keefe is a graduate student in the Department of Computer Science at the University of Minnesota. His research interests include database security and software engineering. He received a BS in electrical engineering and an MS in computer science from the University of Minnesota in 1980 and 1985 respectively.



Bhavani Thuraishingham is a lead engineer at Mitre Corporation, doing research in database security and integrity. Previously she was at Honeywell, Inc., and before that at Control Data Corporation where she worked on database security, knowledge-based systems, and networks. She was also an adjunct professor and member of the graduate faculty in the Department of Computer Science at the University of Minnesota.

Thuraishingham received the BS in mathematics and physics from the University of Sri-Lanka, the MS in computer science from the University of Minnesota, the MS in mathematical logic from the University of Bristol, United Kingdom, and the PhD in recursive functions and computability theory from the University of Wales, Swansea, United Kingdom. She has published more than 30 technical papers on data/knowledge-based systems and computability theory. She is a member of the Computer Society, ACM, and Sigma-Xi.



W. T. Tsai is an assistant professor in the Department of Computer Science at the University of Minnesota. His areas of interest are software engineering, artificial intelligence, computer security, computer systems, and neural networks. Tsai received his SB degree in computer science and engineering from MIT in 1979 and his MS and PhD degrees in computer science from the University of California, Berkeley, in 1982 and 1986, respectively. He is a member of the IEEE.



THE INSTITUTION
OF ELECTRICAL
ENGINEERS

HONG KONG CENTRE

CA—DSP '89

International Advisory Committee

Prof Souguil Ann, Seoul University, Seoul
Prof R. Bogner, Univ. of Adelaide, Adelaide
Prof Zhenming Chai, Academia Sinica, Beijing
Prof Y.T. Chan, Royal Military Coll. of Canada,
Kingston
Prof C.F. Chen, Boston Univ., Boston
Prof A.G. Constantinides, Imperial College, London
Prof E.L. Dagless, Univ. of Bristol, Bristol
Prof Wolfgang Giloi, GMD-TUB FIRST, W. Berlin
Prof H.P. Khincha, Indian Inst. of Science, Bangalore
Dr Paul C.K. Kwok, Univ. of Calgary, Calgary
Prof H.T. Kung, Carnegie Mellon Univ., Pittsburgh
Dr Ju-Hong Lee, National Taiwan Univ., Taipei
Prof H.F. Li, Concordia Univ., Montreal
Prof S.L. Li, Tsinghua Univ., Beijing
Dr C.N. Liu, IBM Watson Research Centre, N.Y.
Prof Phil Mars, Durham Univ., Durham
Prof J.Z. Ou-Yang, South China Univ. of Tech.,
Guangzhou
Prof Yoshiaki Shirai, Osaka Univ., Osaka
Prof Martha Sloan, Michigan Technological Univ.,
Houghton
Prof F.H. Sumner, Univ. of Manchester, Manchester
Prof H.C. Torng, Cornell Univ., N.Y.
Prof Philip Treleaven, University College, London
Prof Jean Vuillemin, INRIA, Paris
Prof Stephen S. Yau, Univ. of Florida, Gainesville
Prof C.K. Yuen, Nat. Univ. of Singapore, Singapore

General Chairman

Prof Tien Chi Chen
Computer Science Department
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
Telex: 50301 CUHK HX
Fax: (852) 0-6954234

Vice Chairman

Dr Francis Chin, University of Hong Kong
Dr S.K. Tso, University of Hong Kong

Secretary

Mr Albert Poon, Hong Kong Government

Technical Programs

Dr Wan-Chi Siu (Chairman)
Hong Kong Polytechnic
Dr W.K. Cham, The Chinese Univ. of Hong Kong
Dr Richard M.M. Chen, Pacific CAD-CAM
Mould Mfg. Co. Ltd., H.K.
Dr Francis C.M. Lau, University of Hong Kong
Dr S.H. Leung, City Polytechnic of Hong Kong

Special Sessions

Dr P.C. Ching (Chairman)
The Chinese Univ. of Hong Kong
Dr C.K. Chan, City Polytechnic of Hong Kong

Local Arrangements

Dr F.C. Chan (Chairman)
China Light & Power Co. Ltd., H.K.
Mr C.M. Mak, China Light & Power Co. Ltd., H.K.

Publications

Mr Albert Poon (Chairman)
Hong Kong Government

Publicity

Dr Y.S. Cheung (Chairman)
University of Hong Kong
Dr C.K. Chan, City Polytechnic of Hong Kong

Steering Committee

Prof Y.C. Cheng, University of Hong Kong
Dr Barry Jefferies, City Polytechnic of Hong Kong
Prof Y.W. Lam, The Chinese Univ. of Hong Kong
Prof W.S. Leung, University of Hong Kong
Dr J.S.L. Wong, Hong Kong Polytechnic

1989 International Symposium on Computer Architecture and Digital Signal Processing

11-14 October, 1989

Hong Kong Convention and Exhibition Centre
Hong Kong

CALL FOR PAPERS

The symposium will bring together researchers, scholars, managers and engineers to share their experiences and new ideas on Computer Architecture and Digital Signal Processing. It will comprise original contributions as well as invited review papers. Topics include but are not limited to the following:

Computer system architecture, concurrent systems, distributed systems, advanced microprocessors, RISC, digital signal processors, graphic processors & systems, cache & memory hierarchies, parallel algorithms, pipelining, vector processors, array processors, systolic arrays, neural networks, interconnections, VLSI computations, delay insensitive systems, architectures for AI, database and special applications.

Filter design, Fourier and other transforms, adaptive algorithms, fast DSP algorithms, deconvolution, multirate processing, multi-dimensional techniques, coding techniques, compression techniques, VLSI for DSP, spectral estimation and modeling, speech processing, image processing, pattern recognition, robotic vision, acoustics, ultrasonics, audio and recording techniques.

Prospective authors are invited to submit four copies of their papers to the Technical Program Chairman:

Dr Wan-Chi Siu
Department of Electronic Engineering
Hong Kong Polytechnic
Hung Hom, Kowloon, Hong Kong
Telex: 38964 POLYX HX
Fax: (852) 3-7643374

Papers must be complete and not exceed 2500 words or 12 double spaced typed pages. Submission of a paper implies a commitment to present the paper if accepted.

Authors' Schedule:

Deadline for submission of papers:

7 April, 1989

Notification of acceptance:

1 July, 1989

Submission of camera-ready papers:

12 August, 1989

STANDARDS

Editor: Helen M. Wood, National Oceanic and Atmospheric Administration, Rm. 1069, Federal Bldg. No. 4, Washington, DC 20233, phone (301) 763-1564.

Extended Industry Standard Architecture design on schedule, 32-bit EISA connector finalized

All key aspects of the Extended Industry Standard Architecture specification—electrical, mechanical, and system configuration details—have been incorporated and distributed to participating developer companies, according to an announcement by the group of major personal computer manufacturers who initiated the EISA

specification last September.

The initial EISA group was formed by AST Research, Compaq Computer, Epson America, Hewlett-Packard, NEC Information Systems, Ing. C. Olivetti & Co., Tandy, Wyse Technology, and Zenith Data Systems.

Finalization of the EISA specification (now over 250 pages in length) has

progressed on schedule, the announcement stated. Utilizing this non-proprietary industry-standard specification, industry vendors have proceeded with development of computer systems, EISA silicon components, add-on boards, and software enhancements. As of January 30, over 100 manufacturers and developers worldwide had purportedly obtained the complete EISA specification with the intent to develop EISA-related products.

Engineering samples of key EISA chip-set silicon components from Intel are expected in the second quarter of 1989, with production shipments to computer manufacturers planned for the second half of the year. Complete PC systems based on EISA, as well as add-on products, are expected to be available by the end of 1989, it was indicated.

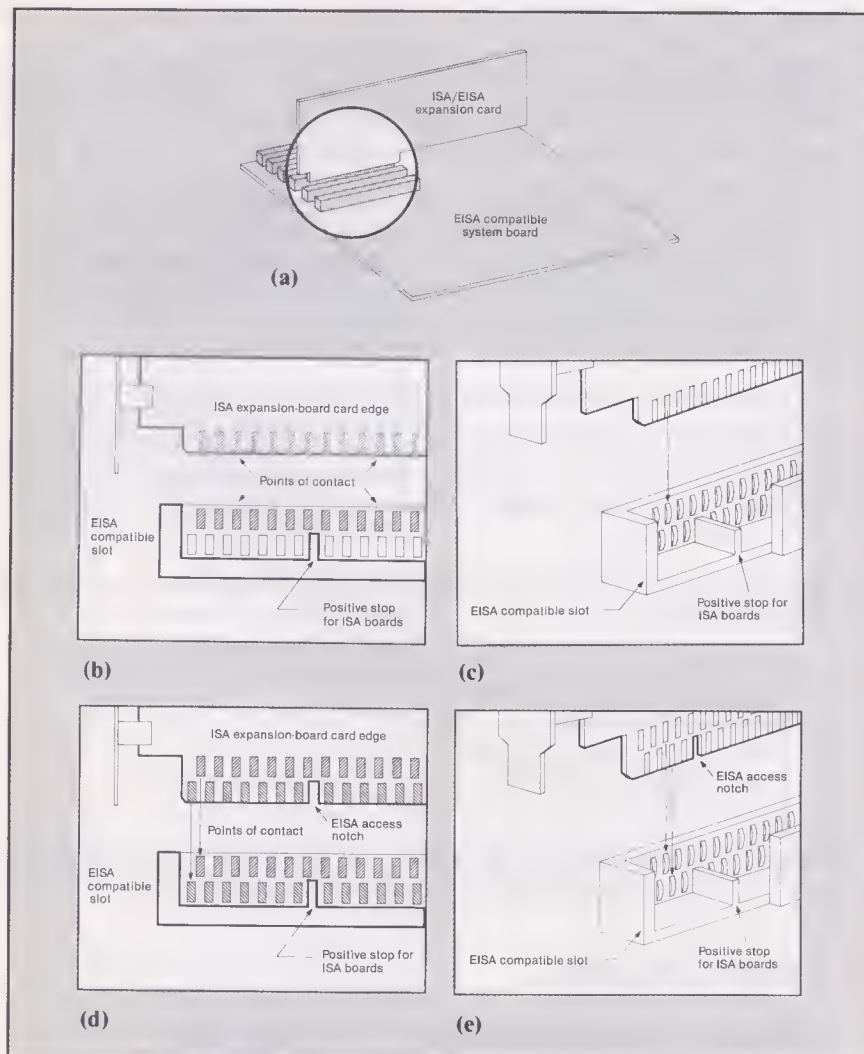
The specification now includes the finalization of mechanical details for the EISA 32-bit connector. The new connector will reputedly allow high-performance 32-bit expansion boards to be installed in PCs utilizing EISA when they become available later this year.

Because the EISA 32-bit connector is the same size as today's industry-standard architecture (ISA) 16-bit connector, it occupies the same amount of system board space. The new connector contains a second row of contacts beneath the standard set of ISA 16-bit contacts to provide 32-bit data and addressing capabilities.

Each 32-bit connector will incorporate physical "stops" that will prevent ISA 8-/16-bit boards from being inserted past the upper row of contacts. New 32-bit EISA expansion boards will be "notched" to bypass the "stops" and allow complete insertion and engagement with the bottom row of contacts.

Burndy Corp., AMP Inc., and other major connector vendors have committed to supplying this component. Production quantities are expected in the second quarter.

EISA specifies a 32-bit bus extension to the existing 8-/16-bit industry standard expansion bus. It delivers 32-bit



The EISA 32-bit connector (a); the ISA card-edge connection (b) and (c); and the EISA card-edge connection (d) and (e).

address and data bus extensions that support memory capacity beyond today's 16-megabyte maximum. It also provides both 32-bit direct-memory access and 32-bit bus master support. With its increased data bandwidth, EISA will reputedly enable PC manufacturers to deliver capabilities supporting future work-group applications while maintaining compatibility with the installed base of PCs and connectibility to a variety of minicomputer and mainframe environments.

Additional features include programmable board setup for automatic configuration of EISA boards and software-aided configuration of existing switch-programmable ISA and future EISA boards, according to the EISA announcement.

Intermetrics develops validation test suite for IEEE standard 1076

Intermetrics has been selected by the VHSIC program office to integrate and disseminate a validation test suite for IEEE Standard VHDL 1076, the company announced. The validation suite will promote uniformity—a critical feature for CAD/CAE tools that implement the standard language.

Uniformity will ensure portability of models across tool sets as well as provide users with a yardstick for evaluating available VHDL tools.

The process of generating a validation test suite is open to all interested parties. Specifically, vendors or others who have language validation tests are encouraged to contribute to the suite. Initially, tests from Intermetrics and Vantage Analysis Systems will form the core of the validation suite.

Tests will be distributed to all interested parties for the cost of the distribution media. Anyone interested in receiving further updates and meeting notices should contact Victor Berman, Intermetrics, 4733 Bethesda Ave., Bethesda, MD 20814, phone (301) 657-3775.

The IEEE Standard 1076-1987 VHDL Language Reference Manual can be obtained by specifying order No. 983 when contacting the Computer Society Press, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-2578, phone (800) CS-BOOKS. In California, dial (714) 821-8380. The IEEE and Computer Society-member price is \$36, and the nonmember price is \$40.

COMPUTER SECURITY SCIENTISTS

Tomorrow's Computing
Technology is Today's Challenge

at
IDA

Some of the nation's most exciting developments in software technology, supercomputer architecture, AI, and expert systems are under scrutiny right now at the Institute for Defense Analyses. IDA is a Federally Funded Research and Development Center serving the Office of the Secretary of Defense, the Joint Chiefs of Staff, Defense Agencies, and other Federal sponsors.

IDA's Computer and Software Engineering Division (CSED) is seeking professional staff members with an in-depth theoretical and practical background in the area of Computer Security. Tasks include efforts on both the design/development of techniques to assess and assure security and providing advice to DoD decision makers on appropriate and feasible policy regarding security.

Specific desired skills and interests include:

- Formal verification, with emphasis on the Ada language
- Secure kernels and reference monitors
- Security in multiprocessor systems
- Fault-tolerance in secure systems
- Operating system, data base and network security criteria
- Testing and evaluation

Specialists in other areas of Computer Science are also sought: **Software Engineers, Distributed Systems, Artificial Intelligence and Expert Systems, and Programming Language Experts.**

We offer career opportunities at many levels of experience. You may be a highly experienced individual able to lead IDA projects and programs . . . or a recent MS/PhD graduate. You can expect a competitive salary, excellent benefits, and a superior professional environment. Equally important, you can expect a role on the leading edge of the state of the art in computing. If this kind of future appeals to you, we urge you to investigate a career with IDA. Please forward your resume to:

**Mr. Thomas J. Shirhall
Manager of Professional Staffing
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311**

An equal opportunity employer.
U.S. Citizenship is required.



Nominations invited for elevating senior members to IEEE fellow grade

Nominations are now being solicited by the Computer Society Fellows Committee to elevate deserving IEEE senior members to the grade of IEEE fellow.

The fellow grade recognizes unusual distinction in the profession and is conferred on a person who has made outstanding individual contributions in

IEEE designated fields. Fellow is the highest grade of membership and the only grade for which members cannot personally apply but must be nominated by another member.

Each candidate must be a senior member when nominated and must normally have been a member in any grade

for five years or more.

In evaluating nominations, the IEEE Fellow Committee considers the following criteria, listed in approximate order of importance:

- (1) Individual contributions as engineer/scientist, technical leader, or educator.

38 Computer Society members designated IEEE fellows

Completing its annual election process for 1988, the IEEE Board of Directors has conferred the title of IEEE fellow on 38 Computer Society members. Of the 392 senior IEEE members recommended to the IEEE Fellow Committee and evaluated, 226 were elected to fellow grade effective January 1, 1989.

The title of fellow recognizes the outstanding contributions of senior members in one or more areas. The alphabetical list below identifies the new IEEE fellows who are Computer Society members and the citation each new fellow received. The IEEE society, group, or council that evaluated the respective candidate appears in parentheses.

Prathima Agrawal, New Providence, New Jersey, for contributions to computer-aided design and testing of integrated circuits (Computer).

Albert P. Albrecht, Alexandria, Virginia, for leadership in the modernization of the National Airspace Traffic Control System (Aerospace and Electronic Systems).

John W. Balde, Flemington, New Jersey, for contributions in the field of electronic packaging and for leadership of industrywide task forces in this area (Components, Hybrids, and Manufacturing Technology).

Lawrence Bernstein, Short Hills, New Jersey, for leadership in the development and incorporation of computer systems in the US national telephone distribution plant (Communications).

Jon T. Butler, Monterey, California, for contributions to the theory and application of multiple-valued logic (Computer).

Bill D. Carroll, Arlington, Texas, for contributions leading to the development of the theory for generalized test functions for combinatorial circuits (Computer).

Ashok K. Chandra, Yorktown Heights, New York, for contributions to the theory of computational complexity, theory of database queries, and VLSI testing (Computer).

Howard Eisner, Potomac, Maryland, for contributions to systems engineering practice and computer aids to that field (Systems, Man, and Cybernetics).

Alexander G. Fraser, Bernardsville, New Jersey, for contributions and leadership in the design of switched virtual circuit networks (Communications).

Hideo Fujiwara, Kawasaki, Japan, for contributions to research and development on testing of large integrated circuits (Computer).

Amrit L. Goel, Fayetteville, New York, for contributions to the reliability of computer software (Computer).

David Kai-Mei Hsiao, Monterey, California, for contributions to research, training, and curriculum development in data modeling, security, engineering, and database systems and machines (Computer).

Aaron Kershenbaum, New City, New York, for contributions to computer and voice communications network design (Communications).

Kane H. Kim, Irvine, California, for contributions to the field of fault-tolerant distributed computing (Computer).

Nathaniel R. Kornfield, Chester, Pennsylvania, for leadership in the development of test technology as a discipline in the design and production of digital computers (Computer).

Paul J. Kuehn, Stuttgart, West Germany, for contributions and leadership in teletraffic theory, communication system modeling, and performance evaluation (Communications).

Richard C.T. Lee, Hsinchu, Taiwan, for contributions to computer engineering education (Computer).

Egon E. Loebner, Palo Alto, California, for contributions to optoelectronic

- (2) Evaluation by an IEEE society.
- (3) Tangible and verifiable evidence of technical accomplishment, such as technical publications, patents, reports (internal or published) made available with descriptions of products, system products and system applications, facilities, and/or services performed.
- (4) Confidential opinions of IEEE fellow references who are qualified to judge the work of the candidate (if possible, these persons should not be associated with the candidate's own organization).
- (5) Service to IEEE (AIEE/IRE) and other professional engineering societies.
- (6) Total years in the profession.

Kits containing this year's nomination forms can be obtained from the Staff Secretary, IEEE Fellow Committee, 345 East 47th St., New York, NY 10017-2394, phone (212) 705-7750; or

from the Member Services Department, IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-2578, phone (714) 821-8380.

The strict deadline for submitting nomination forms B-27, B-3, and B-29 to the IEEE is April 30, 1989.

Tilak Agerwala is again serving as the Computer Society's Fellows Committee chair. He can be contacted at IBM, Old Orchard Rd. Armonk, NY 10504, phone (914) 765-6491.

As indicated in the accompanying article, 38 Computer Society members are newly elected IEEE fellows.

"Clearly, we have many more members who deserve this recognition," Agerwala said. "The society's Fellows Committee urges members to take on the task of identifying worthy candidates, completing the necessary paperwork, and meeting the deadline for

submission of the fellow nominations to the IEEE."

Nominators should start the process early since it takes a fair amount of preparation to make a strong case and identify appropriate references, Agerwala added. A list of Computer Society members who are IEEE fellows can be obtained by contacting Agerwala.

Members are also urged to apply for senior membership since it is one of the requirements for elevation to the fellow grade. Admission or transfer to the grade of senior member requires that the candidate be an engineer, scientist, educator, technical executive, or originator in IEEE-designated fields and active in professional practice for at least 10 years, with significant performance over at least five years.

Applications for senior membership can be obtained by contacting one of the offices listed above or by circling Reader Service No. 204.

materials, devices, and networks, and for transdisciplinary research across life, social, and computer sciences (Electron Devices).

Karl E. Martersteck, Jr., Naperville, Illinois, for technical leadership in the design and development of digital switching systems (Communications).

Gerald M. Masson, Baltimore, Maryland, for contributions in interconnection structures for voice/data communication and fault-tolerant computing (Computer).

Debasis Mitra, Murray Hill, New Jersey, for contributions to the mathematical foundations of computer and communication systems design and analysis (Computer).

Yrjö A. Neuvo, Tampere, Finland, for contributions to digital-signal processing algorithms and engineering education (Circuits and Systems).

Eiichi Ohno, Kanagawa-ken, Japan, for contributions to and leadership in the development of power electronics and industrial control systems using microprocessors (Industrial Electronics).

Daniel L. Ostapko, Mahopac, New York, for contributions to design automation, fault-tolerant computing, and programmable logic arrays (Computer).

Janak H. Patel, Urbana, Illinois, for contributions to the field of computer architecture (Computer).

Madhu V. Pitke, Bombay, India, for technical leadership in the design and implementation of a family of digital switches for India's national telecommunications network (Communications).

Ralph W. Powell, Pittsburgh, Pennsylvania, for contributions to lightning protection of substations, rotating machines, and distribution systems (Power Engineering).

Birendra Prasada, Verdun, Quebec, Canada, for leadership in the development of industry/university research in image coding and processing, and for contributions to digital encoding of image signals (Communications).

Mortimer Rogoff, Washington, DC, for invention of a principal form of spread-spectrum communications and for contributions to the field of electronic navigation systems (Communications).

Alan J. Smith, Berkeley, California, for contributions to the understanding, design, and optimization of computer memory hierarchies (Computer).

Herbert L. Thal, Jr., Wayne, Pennsylvania, for contributions to the application of computer-aided procedures in the

development of microwave components (Microwave Theory and Techniques).

Donald E. Thomas, Jr., Pittsburgh, Pennsylvania, for contributions to automatic design of integrated circuits and systems, and to education in computer engineering (Computer).

Eric A. Vittoz, Cernier, Switzerland, for contributions to the development of micropower integrated circuits (Solid-State Circuits Council).

Thomas W. Williams, Boulder, Colorado, for leadership and contributions to the design for testability (Computer).

Bruce F. Wollenberg, Minnetonka, Minnesota, for contributions to the application of computers in energy management systems (Power Engineering).

William W. Wu, Rockville, Maryland, for contributions to the application of forward acting error coding in satellite communications (Communications).

William A. Wulf, Charlottesville, Virginia, for contributions to computer architecture, design of operating systems, and high-level programming languages and compilers (Computer).

Walter S. Zaengl, Zurich, Switzerland, for contributions to ultrahigh voltage testing and measuring techniques (Power Engineering).

IEEE Educational Activities Board recognizes CSAB-accredited programs

Tom Phillips, Computing Sciences Accreditation Board

The IEEE Educational Activities Board has approved adding computer science programs accredited by the Computing Sciences Accreditation Board to its list of programs with recognized standing.

As a result of action taken at the board's October 1988 meeting, graduates of accredited professional computer science programs now enjoy the same status for membership as graduates of engineering programs accredited

by the Accreditation Board for Engineering and Technology.

Graduation from a program of recognized standing provides the qualifications needed for IEEE membership, an important consideration for persons dedicated to career growth in computer science or engineering. For educational institutions, this provides further evidence of important preprofessional considerations in developing computer-science degree programs.

CSAB is a joint educational venture of the IEEE Computer Society and ACM. Since its inception in late 1984, CSAB has accredited 65 computer science programs at a variety of colleges and universities throughout the US (see p. 65 of the November 1988 issue of *Computer*).

Well qualified volunteers from the society and the ACM form CSAB's corps of 175 trained visiting team leaders and program evaluators.

Team from consortium wins 1988 Gordon Bell Award

The team of Phuong Vu, Horst Simon, Cleve Ashcraft, Roger Grimes, John Lewis, and Barry Peyton has been named the winner of the second annual Gordon Bell Award. Ardent Computer's Gordon Bell is sponsoring the 10-year series of awards to promote parallel computing.

The winning team, representing a consortium from Cray Research, the National Aeronautics and Space Administration, Yale University, Boeing Computer Services, and the Oak Ridge National Laboratory, will receive \$1,000 for its work on software for sparse matrix factorization.

Ted Lewis, editor-in-chief of *IEEE Software* which administers the judging and presentations, said the 1988 winners were being recognized for "outstanding achievement . . . in the category of raw performance—the fastest computer on Earth." He said the members of the team "are to be congratulated for their excellent work in parallel computing."

The panel of judges that selects the Bell Award honorees did not designate a second prize winner this year because the panelists felt no entry met the requirements in either the best price/performance or the best compiler parallelization categories.

Two entries merited honorable mention, however. Richard Pelz of Rutgers University was singled out for a hypercube implementation of the Fourier pseudospectral method, and Marina Chen, Young-il Choo, Erik DeBenedictis, Jingke Li, and Janet Wu of Yale University were recognized for their work on a financial application written in Crystal and compiled for two hypercube multiprocessors.

The formal presentations will take

place during Compcon Spring 89, February 27-March 3 in San Francisco.

Shreekanth Thakkar of Sequent Computer Systems coordinated the panel of judges, led by Alan Karp of IBM Palo Alto Scientific Center. David Kuck of the University of Illinois Center for Supercomputing Research and Develop-

ment, Jim Browne of the University of Texas at Austin, Jack Dongarra of Argonne National Laboratory, and Ken Kennedy of Rice University assisted Karp in sifting through the entries.

IEEE Software will publish a detailed technical analysis of the winning entries in its May issue.

TAB setting up TC dues-collecting procedures

Laurel V. Kaleda, Second Vice President, Technical Activities

The Computer Society Technical Activities Board is in the process of establishing procedures to be used to allow the various technical committees to begin collecting dues for membership (see p. 104 of the January 1989 issue of *Computer*).

Since TC dues collection will be quite a departure from the past, members of TAB want to make sure this policy change is implemented properly.

For a long time, a number of the TCs have wanted to extend their activities but have felt constrained by the need for a reliable source of funds to support such additional events. Membership dues specifically targeted to a TC will enhance that TC's pool of funds for its activities.

In general terms, any TC wishing to charge dues for membership will propose the fee and detail the TC activities the dues will enable the TC to offer its members. The TC will present its plan as a total program to be reviewed by a TAB committee set up for this purpose.

The subcommittee will be responsible

for ensuring that each TC allowed to charge dues is, in fact, offering a reasonable package of services for the fee. Put another way, TAB members want to ensure that those who pay TC membership dues get their money's worth.

TAB is very interested in the opinions of society members on this proposal. It's an entirely new direction for our TCs but one that seems necessary to allow them to initiate new programs with increased technical services, without requiring additional Computer Society member dues. We members of TAB hope that you agree.

Please let us know your feelings on this matter in either or both of the following ways:

(1) Circle 185 on the Reader Service Card if you favor the concept of the TCs charging dues to expand their activities, or circle 186 if you're against the TCs collecting dues.

(2) Send your comments on this issue to Laurel V. Kaleda, IBM, L43/098, 5600 Cottle Rd., San Jose, CA 95193.

UPDATE

Contributions to Update are welcome. Send news of industrial or university research and of public policy or professional issues to Update Editor, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, or to Bruce Shriver, Editor-in-Chief, Dept. of Decision Sciences, University of Hawaii, 2404 Maile Way, Honolulu, HI 96822.

NSF seeks 14 percent boost in 1990 budget

National Science Foundation funding for networking and communications research would more than double in 1990 under the proposed NSF budget submitted to Congress February 9 as part of President George Bush's budget package. The Computer and Information Science and Engineering category would receive a 25.7 percent funding boost if the entire NSF budget were approved.

The total NSF budget request for 1990 is \$2.15 billion, 14 percent above the 1989 appropriation of \$1.89 billion. The proposed budget is in line with the agency's plan to double its budget over a five-year period that began with fiscal 1989.

Within Computer and Information Science and Engineering, Networking and Communications Research and Infrastructure would receive \$27.133 million, a 53.4 percent increase over 1989's \$17.683 million, which in turn was more than double the 1988 apportionment. Computer and Computation Research would receive \$23.673 million, an 11.9 percent increase; Information Robotics and Intelligent Systems would receive \$21.637 million, a 13.2 percent increase; Microelectronic Information Processing Systems would receive \$18.952 million, a 26.7 percent increase; Advanced Scientific Computing would receive \$71.689 million, a 28.4 percent increase; Cross-Disciplinary Activities would receive \$21.416 million, a 23 percent increase; and the related Science and Technology Centers would receive \$6.7 million, a 13.1 percent increase.

Overall, individual research projects and facilities would receive \$130 million more than in 1989, representing about \$1.32 billion or 60 percent of the proposed budget. In addition to high-performance computing, NSF is requesting increased funding for mathematics, materials research, global change, and biodiversity. Disciplinary research accounted for about 83 percent of the 1989 budget.

NSF is also proposing a \$15 million increase to upgrade hardware for NSF's National Supercomputer Centers, \$6 million for NSFnet, and a doubling of funds

for materials research facilities, including a proposed high-magnetic-field laboratory to be used by materials scientists as well as physical and biological scientists.

Research centers and group research would receive more than \$376 million, an increase of \$56 million over the current budget. This category includes cooperative efforts by scientists from different disciplines or specialties and the sharing of large-scale equipment or facilities.

NSF plans to continue supporting the 11 Science and Technology Centers established in 1989. The requested \$20 million increase in this area would fund 10 to 12 additional centers. Total center funding for 1990 would be \$47.5 million.

Between two and four new Engineering Research Centers would be established under the proposed budget. The request for all ERCs is \$45.7 million. Between eight and 11 new Industry/University Cooperative Research Centers would be established and the I/UCRC budget would increase to \$5 million.

The budget for education and human resources would grow to \$347 million, an increase of \$63 million. Pre-college-level funding would increase by \$10 million to \$129 million for programs including teacher training and continuing education, the development of instructional materials, museum exhibits and television programs, and the Young Scholars Program.

Bitnet members approve merger with CSnet

Galen Gruman, IEEE Software

Members of the Bitnet research computer network voted 195-21 Feb. 6 to approve a proposed merger with the CSnet network. The vote was "a strong mandate," said Jim Conklin, director of Bitnet's Network Information Center in Washington, D.C. Bitnet and CSnet are now working on a detailed proposal — including a transition plan and articles of incorporation — that CSnet's parent company, the University Corp. for Atmospheric Research, will vote on in early summer.

UCAR's approval is expected, although some board members are concerned about the \$500,000 that an independent consulting firm said the merger would cost the two networks for meeting and legal expenses. Both Bitnet and CSnet spokesmen said the consultant's estimate was much too high.

While spokesmen for both organizations say the networks are financially healthy, "we felt [that] as the [National Science Foundation-funded] NSFnet and regional networks continued to grow, there wasn't a lot of room for two mail-

transfer and file servers at the low end of the spectrum," Conklin said. "We'd be stronger [merged] than if we stayed separate," he said, because "the two networks — although their technologies are different — are very similar in their goals. Both are oriented towards low-cost academic services," he said.

Even with the growth in regional networks and national networks like NSFnet, "we have a niche to fill that can't be covered by the midlevel networks: industry and small schools," said Dan Ruttenberg, CSnet's executive director. The merger of CSnet and Bitnet would position them to offer "affordable connectivity to Internet and provide a high level of service to new and small member institutions," he said.

The merger would give CSnet "the ability to combine with a much broader user base and provide a broader connectivity with computer-science departments and industry with the rest of academia," Ruttenberg said. "It would [also] give us more clout as a substantial component of NSFnet," he said.

If UCAR approves the merger, users will see little change for the first few years, Conklin said. "In the short term, we'd offer the separate offerings that we're now offering," he said. Ruttenberg said he thinks the operations would merge more quickly, perhaps a year after the new network's parent organization is established.

Bitnet members would gain access to a higher bandwidth system. The limited bandwidth of Bitnet's current links has resulted in increasingly aggravating bottlenecks, Conklin said. In the long term, Bitnet users would gain access to CSnet's TCP/IP communications protocol, which provides higher bandwidth and more capabilities than the aging RSCS protocol Bitnet uses, Conklin said.

Bitnet has already implemented a project called Bitnet 2 to let the RSCS protocol work over TCP/IP links. That would make the network merger more transparent to current Bitnet users — especially to European users, who tend to use RSCS, Conklin said.

The merger proposal has raised questions among some Bitnet members about a potential loss of financial independence, Conklin said. Bitnet uses leased lines paid for by member institutions who also buy their hardware, which gives the network independence from the federal budget uncertainties that users of federally sponsored networks like NSFnet might face, he said. As a government-supported system, CSnet is vulnerable to budget swings and some Bitnet

members feared the merged network might also be vulnerable, Conklin said. However, the grass-roots support from universities and companies that would use the network in so many states would most likely ensure that Congress appropriate sufficient and stable funds, Ruttenberg said.

About half of CSnet's members are also institutionally members of Bitnet, but the CSnet membership is oriented more toward academic departments and Bitnet's more toward an organization's central computing departments, Conklin said, so he expects that "a significant number of organizations will remain on both." Ruttenberg concurred.

News briefs

Microcode protected. Microcode is a computer program and therefore protected under copyright laws, US District Court Judge William F. Gray ruled February 7. The ruling came at the conclusion of a 4½-year court battle in which Intel claimed that NEC's V-series microprocessors violated the copyright on Intel's 8086/88 microcode.

Although he decided that microcode is protected, Gray ruled in NEC's favor in the main dispute, finding that Intel forfeited its copyright by allowing copies of the 8086/88 chip to be distributed without copyright notice.

US adheres to Berne convention. The US began following the Berne copyright convention March 1, 1989. Congress ratified US acceptance of the international copyright treaty in October ("US Accepts Berne Copyright Convention," *Soft News, IEEE Software*, Jan. 1989, p. 118).

Under previous law, a work had to display the © symbol, the abbreviation "Copr" or the word "copyright" followed by the year of first publication and the owner's name. Works published after March 1, 1989, will be protected whether or not they bear the notice.

However, the Copyright Office still urges people to display a copyright notice, said Richard Glasgow, the office's assistant counsel general. "If it is not used, an infringer can use the defense that he is an innocent infringer" because there was no notice of copyright ownership to tell him he was infringing it, Glasgow said. The maximum damage award that you can collect for willful infringement is \$100,000 — five times that for unknowing infringement. The new law doubled the maximum damages so people would have more incentive to display the copyright notice, Glasgow said.

Scientists & Engineers

SOLVE REAL-WORLD PROBLEMS FROM FIRST PRINCIPLES

Few challenges today come with ready-made answers. Add the complexity of high technology and the possibilities multiply. As the central research facility for Martin Marietta Corporation, Martin Marietta Laboratories is solving some of the toughest problems around in Information Technology.

Opportunities are available for:

MANAGER—AI

PhD with 8-10 years experience in AI (image understanding, pattern recognition, sensor fusion, parallel processing, and/or neural networks) to manage sensor interpretation and planning/assessment research efforts. Must have previous experience as a group leader and possess outstanding communication skills including professional publications.

KNOWLEDGE ENGINEER—AI

MS with minimum 2 years expert system applications background (modeling and simulation experience preferred) to lead research project in computer system risk assessment. Must have developed at least one medium to large expert system and possess excellent analytical skills.

ENGINEER—AI

BS/MS with 3 years experience in simulation & graphics to support research in process modeling and control, sensor interpretation, AI, and machine learning. Expertise should include modeling and simulation, process control, image processing, statistical pattern recognition, and decision theory. "UNIX" and "C" required.

SCIENTIST—NEURAL NETS

MS/PhD to support research in adaptive learning and development of neural network models for vision, automatic target recognition, sensor interpretation and fusion, and adaptive control. Requires 2-3 years experience in algorithm development and software simulation. Parallel (especially systolic array) processing experience a plus.

SCIENTIST—DIGITAL COMMUNICATIONS

MS/PhD with 3-5 years experience in communications systems/network architectures, network analysis and modeling, and ISDN to conduct reconfigurable communications network research, design simulation models, and develop network management and control systems. FORTRAN required, knowledge of network simulation software is a plus.

OPTICAL SIGNAL PROCESSING

MS/PhD needed to design and build a state-of-the-art hybrid optical/digital signal processing system and assist in hardware development. Must have experience in optics, signal processing, information processing, and pattern recognition.

For consideration, send resume to:
Beth Jones, Personnel Recruiter,
MARTIN MARIETTA LABORATORIES,
1450 South Rolling Road, Baltimore, MD
21227. (301) 247-0700 x2331.

An equal opportunity employer
U.S. citizenship required.

MARTIN MARIETTA

MARTIN MARIETTA LABORATORIES



Supercomputing '89

November 13-17, 1989
Reno/Sparks Convention Center, Reno, Nevada

CALL FOR PARTICIPATION

Supercomputing '89 continues the tradition established at the '88 Conference and will bring together supercomputing system researchers, designers, managers, and computational scientists and engineers to report advances and experiences, state needs, suggest future directions and exchange information. It will include a technical program of invited and contributed papers, tutorials, poster sessions, vendor and research exhibits, and product briefings and demonstrations.

Papers

Authors are invited to submit papers which report concrete results and experiences. Papers reporting important negative results are also encouraged. Submit five copies to the Program Chairperson by May 1, 1989.

Posters

Authors who prefer an informal environment that fosters interaction with the conference attendees are encouraged to submit poster proposals. Submit five copies of the poster proposal to the Program Chairperson by June 1, 1989.

Vendor Exhibits

An opportunity exists for vendors to exhibit their supercomputing technology during three days of Supercomputing '89. Interested parties should contact the Exhibits Chairperson as soon as possible to arrange for floor space.

Research Exhibits

A limited amount of space will be set aside at Supercomputing '89 to allow researchers to set up and exhibit or demonstrate their work. Submit a brief proposal to the Exhibits Chairperson as soon as possible, but not later than March 31, 1989.

Tutorials

The traditional half-day or full-day lectures will be presented, with view-graphs distributed to attendees. Submit proposal by February 28, 1989 to Tutorials Chairperson.

Extended Tutorials

These sessions provide an opportunity for a mini-course introduction to a topic. The course begins before the conference with advanced mailing of course material. Submit proposal by February 28, 1989 to Tutorials Chairperson.

Workshops

Workshops are of the "birds-of-a-feather" variety where registrants participate in an interactive workshop environment. The organizer is a participant/facilitator rather than a lecturer. Submit proposal by April 30, 1989 to Tutorials Chairperson.

Contacts:

General Chairperson	Program Chairperson	Tutorials Chairperson	Exhibits Chairperson
F. Ron Bailey	Gary Johnson	John Riganati	Howard Johnson
Mail Stop 258-5	North Carolina Supercomputing Center	Supercomputing Research Center	15694 East Chenango
NASA Ames Research Center	P.O. Box 12732	4380 Forbes Boulevard	Aurora, CO 80015
Moffett Field, CA 94035	Research Triangle Park, NC 27709	Lanham, MD 20706	303/693-8291
415/694-4500	919/248-1825	301/731-3741	howardj%csugreen.bitnet
rbailey@prandtl.nas.nasa.gov	garyj@mcnc.org	riganati%super.org@sh.cs.net	@cunyvm.cuny.edu

Sponsored by : ACM SIGARCH and the Computer Society of the IEEE

In cooperation with: Argonne National Laboratory • Lawrence Livermore National Laboratory • Los Alamos National Laboratory
• NASA Ames Research Center • National Center for Atmospheric Research • National Science Foundation • SIAM Activity Group
on Supercomputing • Supercomputing Research Center

CALL FOR PAPERS

IEEE INTERNATIONAL WORKSHOP ON TOOLS FOR AI

"Architectures, Languages and Algorithms"

October 23 - 25, 1989

GEORGE MASON University

Fairfax, VA 22030



Sponsored by: IEEE COMPUTER SOCIETY

In Cooperation with the following IEEE Computer Society Technical Committees: Pattern Analysis and Machine Intelligence TC, Computer Languages TC, Computer Architectures TC, Software Engineering TC, Data Engineering TC.

In Cooperation with: E-Systems Corp.



SCOPE : The goal of this workshop is to provide the ground for the development of general and/or special purpose tools, such as architectures, languages, algorithms and DBs for AI. We expect the contributors in AI areas to share knowledge and experience highly beneficial for the great variety of attendants and for the future research efforts.

COMMITTEES

Steering

C.V. Ramamoorthy, UC, Berkeley, Co-Chair

N.G. Bourbakis, George Mason Univ.

B.W.Wah, Univ. of Illinois, Urbana

W.T. Tsai, Univ. of Minnesota

S. Emmett, E-Systems Corp.

Program

M. Papazoglou, GMD, Germany, Co-Chair

A. Klinger, UCLA

P. Shue, Purdue Univ.

T. Selis, Univ. of Maryland

H. Stephanou, George Mason Univ.

J.C. Latombe, Stanford Univ.

R. Newcomb, Univ. of Maryland

A. Kent, Univ. of Pittsburgh

S.Y. Kung, Princeton Univ.

J.K. Aggarwal, Univ. of Texas

R.S. Michalski, George Mason Univ.

P.S. Wang, Northeastern Univ.

L. Kerschberg, George Mason Univ.

H. Freeman, Rutgers Univ.

K.H. Kim, UC, Irvine

A. Dollas, Duke Univ.

P. Ligomenides, Univ. of Maryland

M. Squires, TRW

S. Barry, TRW

Plenary Speakers

A.P. Sage, George Mason Univ.

C.V. Ramamoorthy, UC, Berkeley

Deadlines:

June 15, 1989, submission

July 15, 1989, acceptance

August 15, 1989, Final papers

TOPICS

Intelligent Architectures, Languages & Algorithms on

- Vision
- Speech
- Text
- Computations
- Neural Nets
- Learning Schemes
- Knowledge Engineering
- Expert Systems
- Databases
- Robotics
- Applications
- VLSI & CAD

Proposals for high quality Invited sessions are welcome.

Submit four copies of a full paper of 5000 words or a summary of 1200 words, by June 15, 1989. The best articles will be considered for publication in Int. Journals related to AI (special issue). In addition, an award will be given to the best student's paper of the Workshop.

Submit papers and proposals to:

Professor N.G. Bourbakis, General Chair
George Mason University
Dept. of Electrical & Computer Engr.
Machine Vision Lab
4400 University Drive
Fairfax, VA 22030 tel.(703)4253930

PRODUCT REVIEWS

Editor: Richard Eckhouse, MOCO, Inc., PO Box A, 91 Surfside Rd., Scituate, MA 02055, Compmail+, r.eckhouse

Data control, acquisition, analysis, and display

Richard Eckhouse, Product Reviews Editor

This month we focus on hardware and software for the control, acquisition, analysis, and display of analog and digital data. Engineers and scientists utilize both individual components and complete systems for such tasks as data capture, waveform analysis, process control, pattern recognition, and testing. Because of the broad range of needs and uses, I have purposely covered an equally broad range of products.

I would like to thank Michael Dediu for his review of Interactive Laboratory System Version 6.0, which I have included.

Asyst

Asyst version 2.1 from Asyst Technologies is probably the most difficult review I've ever had to write. By that statement I don't mean the package is hard to use, poorly documented, or of little value. Quite the contrary. The difficulty is taking in the sheer magnitude of this impressive system, from its five-manual set to the sophisticated examples that demonstrate the power and flexibility of the system.

Just what is Asyst? Before I started this review, I would have described it as a collection of data acquisition, analysis, and display routines to make it easier for the PC user to automate those processes within the laboratory environment. Now, however, I am almost at a loss for words to adequately describe just what Asyst can do — hence my opening statement that writing this review proved to be most challenging.

Let's drop back a bit and focus on what you might expect from a package like Asyst. Typically, a user has a need to perform an experiment, control a process, or analyze and display some collection of data. While most programming languages allow you to write the necessary code to perform any of these tasks, the thought of doing so is enough to stop you before you start. After all, in most cases these tasks are one of a kind, and

the effort that goes into developing a complete solution generally far exceeds the time required to gather, compute, and display the results. What is called for, then, is a process that speeds up the underlying steps (without complicating the process) so that the total time required is in proportion to the value of the results desired.

The interesting thing about Asyst is that it not only supports this process but also allows you to embellish the results so that they appear every bit as personal and professional as if they were independently developed. Thus, at one extreme, the novice user can — quickly and simply — interactively execute the necessary steps to acquire, analyze, or display experimental data. At the other extreme, the sophisticated user can customize the process to control compiled programs with keystrokes and function keys using fancy graphics, as well as stepping through a large collection of routines that perform all the necessary operations for rather complicated experiments.

What is Asyst?

Asyst is a scientific software package that runs on an IBM PC, XT, AT, PS/2, or compatible and turns it into a workstation that allows you to reduce data while collecting it and graph it as you proceed. For me to explain all of this, you have to understand how Asyst really works.

In actuality, Asyst consists of a stack-oriented programming system with a core made up of a large collection of command words. A large but basic set of these commands comes with the base system. Additional system command words are stored in precompiled overlay files that you can add to the base set as part of the configuration process to create new working modules. Most users will create multiple working modules to account for the different and independent activities they require. One such module might be for creating programs, another for acquisition, and a combined one for

analysis and display.

In addition to specifying what modules to assemble into a working set, configuration also allows you to specify the hardware, the memory layout, graphics and text colors, display modes, and the characteristics of analog/digital plug-in boards. The saved system then becomes the working module.

Typically, the new user will go through the separate *Up and Running with Asyst 2.0* manual to get a flavor of what this system will do. This extremely thorough and well done manual covers every aspect of Asyst and prepares you for the more voluminous documentation that covers the four modules making up Asyst: (1) System/Graphics/Statistics, (2) Analysis, (3) Data Acquisition, and (4) GPIB/IEEE-488 Interface.

What you need and what you get from this start-up manual is how Asyst works. You quickly learn how to execute simple commands to push and pop elements on the stack, that there are really two stacks for numbers and strings, and that the normal arithmetic operators work equally well on scalars and arrays. While the Reverse Polish Notation required to execute stack operations might take some getting used to, it really should be familiar to most engineers and computer scientists. And the efficiency it provides in generating good programs far outweighs any difficulties the inexperienced user might have at first.

What takes some time is learning all the new command words that provide the system's functionality. The English-language-like names for these commands greatly aid the process. Thus, you find yourself quickly graduating from multi-liners like

```
27
SQRT
5.4
+
VOLTS
:=
```

where SQRT is a built-in word and VOLTS is a user-defined scalar, to one-

liners like

```
27 SQRT 5.4 + VOLTS :=
```

(which takes the square root of 27, adds 5.4, and assigns the value to the variable VOLTS). Next you move on to more complicated "colon definitions" that give you the ability to create your own commands. To write a new command "ASSIGN.VOLTS" that would add the last expression to the set of words, you would write

```
: ASSIGN.VOLTS 27 SQRT 5.4 +  
VOLTS := ;
```

While these examples are very simple, much more complicated sequences would lead to rather sophisticated commands that could extend the system or provide unique capabilities as needed by a particular application. Indeed, many such programs come on the demo diskette and are discussed in great detail in both the *Up and Running* manual as well as the manuals covering a particular module.

Note also that, if you save the working system with the programs you wrote yourself, they become part of the larger working system, which you can reload at a later time. Alternatively, you can compile and load your programs as overlays. In fact, any of the system overlays not part of the working system can be loaded at any time. More sophisticated programs can thus load and reload many different overlays, depending on the section of a program being executed, thus conserving memory.

As most PC users know, memory is still a precious commodity. Programs do grow large, and Asyst's overlay structure helps solve the problem. Still, the segmented addressing of the PC becomes a limiting factor in Asyst, where the number of data points is limited to one segment. This translates into 32 Kbytes worth of single-precision integers, 16 Kbytes of reals, and so on. Of course, using the Asyst file structure can make this more palatable, particularly since Asyst provides the full set of DOS-like file operations. Not only that, but you can read Basic and Lotus 1-2-3 files, as well as read or write random binary files. The only catch is that you can only open one file at a time for reading or writing.

An extensive system such as Asyst requires an extensive help system. Called up either from the command line with the word "help" or by pressing the F1 key, the help command provides multiple levels of assistance, from useful descriptions of the Asyst system to context-sensitive help about particular commands. The command-line trace feature is part of

the help system. When you create a new command that results in an error and hence needs debugging, the trace feature provides a list of the words used prior to the occurrence of the error message.

While not as detailed as you might like, this feature is sufficient to help you locate the proximity of the error condition.

Asyst's text editor facilitates writing programs. You could use any editor capable of manipulating ASCII files, but the one provided with the Asyst system has the advantage that you don't have to leave Asyst to use it. In addition, when you exit the editor and load a program containing an error, you can call back the editor at the place where the error occurred, correct the problem, and continue with the program load. For long programs with multiple errors, this saves a lot of time.

I especially liked the text editor's simplicity. There are just enough function key commands to do what needs to be done, and there are no special key sequences like in Wordstar.

In addition to the text editor, another editor, called the array editor, is equally helpful. By laying out the data in rows and columns, this editor makes it easy to spot errors in the array data and to correct them. Again, a simple command structure makes the editing process quite straightforward. The one catch is that the editor only works for one- and two-dimensional data sets.

I did not like a few things in Asyst. First, array indices always start with 1 (that is, no negative or zeroth element). Second, the manuals, while very good tutorials, occasionally use a word not defined in the body of the chapter. Fortunately, an excellent index ties into a very complete glossary of all the words and nomenclature, so looking up the new word is pretty easy. Last, some things just aren't covered. For example, I had to guess how to create a stand-alone working system so that, when I called it up from DOS, it automatically started execution of my working Asyst system. Fortunately, I once again had an alternative to finding answers — using the free (but not toll-free) hot line.

The details

As I mentioned, Asyst comes in four modules. The base price of \$1,695 includes modules 1 and 2, System/Graphics/Statistics and Analysis. Adding either module 3 (Data Acquisition) or module 4 (GPIO/IEEE-488 Interface) adds another \$400 to the price. If you purchase both, the total price is \$2,295 and you save \$200. The important point here is that the base system can be used alone — you

can write or purchase a separate acquisition module or simply use Asyst to analyze and display data generated by whatever means, including spreadsheets, programs, pencil/paper studies, etc.

The System/Graphics/Statistics manual set consists of two volumes. The first volume is tutorial and takes you through the underlying concepts; mathematical functions; stack operations; scalars, complex numbers, and arrays; graphics and data scrolling; strings; statistics; program generation; conditionals and loops; files; system utilities; and using RS-232-based devices. The second volume explains the conventions used and contains a complete glossary of all the Asyst commands. The appendixes include a revision history, how to install a plotter, data file structure, and error messages.

The Analysis manual describes the various system overlays for evaluating polynomials; vectors and matrices; eigenvalues and eigenvectors; least square approximations; Fourier transforms; curve fitting; analysis of variance; and axonometric and contour plots. The Acquisition manual covers both nonbuffered and buffered A/D, D/A, and digital I/O, time synchronization, foreground/background operation, and high-performance analog I/O.

The GPIB/IEEE-488 Interface manual covers the IEEE-488 standard; controller functions and bus commands; buffered communications; asynchronous communications; service requests and polling; and an example system.

Each manual includes numerous appendixes, and each comes with a complete index into all the Asyst manuals. Each is very well written, with a tutorial style that makes them easy to read and understand.

In addition to the manual set, the complete Asyst software system comes on seven floppy disks. The package also includes a copy protection dongle that plugs into a parallel port. Besides a PC, you will need a minimum of 512 Kbytes (640 Kbytes recommended), two floppy drives (but a hard drive is really a necessity), a graphics display adapter (EGA is best), and a math coprocessor. Asyst can also make use of expanded memory by using "tokens" that hold the array values in such memory.

The acquisitions module supports a wide variety of interface boards. Familiar vendors include Analog Devices, Data Translation, IBM, Keithley, MetraByte, and Techmar. During my evaluation I used an Analog Devices RTI-815 board installed in a Compaq Deskpro 286 running at 8 MHz with 1 Mbyte of memory and 2 Mbytes of expanded memory, plus a NEC EGA display adapter and multisynch monitor.

The specifics

The Analysis module is an integral part of the overall Asyst package. On the surface it can be likened to any other set of subroutines for the manipulation and reduction of scientific and engineering data. It includes the usual set of numerical analysis routines, including differentiation, integration, solutions for simultaneous equations, curve fitting, frequency analysis, and analysis of variance, along with some interesting routines for 3D graphics.

However, like all the other parts of this package, the sum is greater than the individual parts. Here is where Asyst's analogy to "words" really takes on meaning. Just as it is true that the meaning of a sentence is much more than the individual meanings of the words that make it up, in the case of Asyst, the programs are much more complex than the individual and simpler words that make them up. (See the accompanying sidebar for a sample program that I wrote to give you a flavor of the language.)

As I've said, there is much more to Asyst. For instance, not only can you apply the standard arithmetic operators to arrays just as you would to scalars, you get specific array operators for inner and outer products; reshaping, rotating, extracting, and looking up subsets of values; and concatenating or laminating arrays together. The graphics capabilities include bar, pie, and scatter charts, fast graphics redraw using multiple graphics memory pages, and support for dual monitor systems. In addition, Asyst includes a data scroller so that you can view large data sets by windowing into sections of interest. Finally, a set of words handles the PC's RS-232 COM ports in both polling and interrupt modes.

The Acquisition module is worth special mention. Here's where Asyst really outshines the competition because it provides both buffered and nonbuffered data acquisition and control either in polled or DMA mode. Spooled disk data rates to as high as 45 kHz for an 8 MHz AT are possible, and 120 kHz is claimed using an EMS RAM disk. But, for the majority of users, simple single or double buffering is sufficient. Here again, Asyst's many commands make this a simple matter.

I used the demonstration programs with an Analog Devices RTI-815 board. It was possible to gather data in bursts, to files, with the results presented on the monitor, and in foreground/background mode. Some limitations exist due to (I) the segmented address space of the PC that restricts maximum buffer size to

64 Kbytes, (2) the PC's clock rate, and (3) the slowness of the PC's display. In all cases, alternatives exist, from using DMA data gathering to installing a Dataq waveform scroller board. Thus, the Asyst package once again provided the versatility and functionality lacking in most other systems.

Recommendation

Asyst is a full-featured system with rich, powerful commands that allow you to rapidly create sophisticated programs for the acquisition, analysis, and display of real-time and experimental data. It

Sample Asyst program

I've written a short program that generates a square wave (by creating a sine wave and then taking the SGN function of it), then performs an FFT on it. While the RPN notation might be a bit unfamiliar, I think the program still shows off many of the features of Asyst, including the use of the stack, graphics, and loops, and the overall nesting of subroutines defining new words. Of course, this simple example really doesn't exploit much of what Asyst has to offer, but it does give the flavor of the language.

```
REAL SCALAR #CYCLES      \ Create real scalar number of cycles
VUPORT ONE                \ Declare two viewports
VUPORT TWO

: INIT.SCREEN             \ Create new word to set up the screen
  GRAPHICS.DISPLAY        \ Initialize graphics display device
  21 0 24 79 SET.WINDOW   \ Make text window at bottom
;

: GET.VALUE               \ Create new word to get a number
  CR ." Number of cycles: " \ Ask for the number of cycles
  BEGIN                  \ Start repeat loop
    #INPUT NOT           \ Input a number and force stack result false
    WHILE                \ If stack holds true, then non-number entered
      CR ." Invalid reenter: " \ Put out error message
    REPEAT               \ Repeat loop until false (e.g., number entered)
;

: MAKE.SQUARE             \ Create new word to make a square wave
  RAD                    \ Units are radians
  256 REAL RAMP           \ Make a 256 point ramp from 1 to 256
  1 - 256 / 2 * PI *      \ Reset so 0 to 255 and convert to radians
  #CYCLES * SIN SGN       \ Multiply by no. cycles, take SIN, and square
;

: SETUP.VUPORTS           \ Create new word to setup viewports
  ONE                    \ Window ONE
    0.0 0.6 VUPORT.ORIG   \ Set origin in upper half of screen
    1.0 0.4 VUPORT.SIZE   \ Set size to fit in upper half
  TWO                    \ Window TWO
    0.0 0.2 VUPORT.ORIG   \ Set origin in lower half of screen
    1.0 0.4 VUPORT.SIZE   \ Set size to fit in lower half
;

: COMPUTE.AND.PLOT        \ Create new word to generate and plot results
  SETUP.VUPORTS          \ Setup the viewports as defined
  DUP FFT ZMAG            \ Copy sq. wave on stack, take magnitude of FFT
  SUB [ 1, 129 ]          \ Consider only half of FFT results
  TWO Y.AUTO.PLOT        \ Plot lower viewport from stack
  ONE Y.AUTO.PLOT        \ Plot upper viewport from stack
;

: SHOW.FFT               \ Create new word to show the FFT
  INIT.SCREEN            \ Initialize screen windows
  GET.VALUE              \ Get the number of cycles to plot and transform
  #CYCLES :=             \ Pop value off stack into integer variable
  MAKE.SQUARE            \ Make square wave
  COMPUTE.AND.PLOT       \ Plot both FFT and square wave
;
```

stands alone in the class of such systems because of its versatility. New users will need to spend a fair amount of time just to learn the syntax, semantics, configuration, and operation of this system. In fact, I believe the learning process is a

continual one, where even experienced users will find new features and ways to use the system to enhance their particular applications. While Asyst is a somewhat expensive system with a base price of \$1,695, its capabilities and versatility

make it worth this price just in time saved.

Contact Asyst Software Technologies, 100 Corporate Woods, Rochester, NY 14623, phone (716) 272-0070.

Reader Service 20

Laboratory instrumentation using board-level devices

The GPIB (or General Purpose Interface Bus) was developed as a flexible way to interface laboratory and test instruments to digital computers. Known also as the IEEE-488/GPIB, this standard has become a flexible, practical, and popular way to connect computer-controlled instrumentation to a wide variety of machines. Using the standard entails having a device with an IEEE-488 interface built-in or as an extra cost option, a GPIB cable, and an interface board that plugs into a PC.

Recently, MetraByte introduced the Personal Computer Instrument Products (PCIP) as a less costly alternative to the standard. Their objective was to offer a set of boards that would plug directly into a PC bus and would use the PC's display as an alternative to the traditional analog or digital display common to most devices. Equally important was that each of the MetraByte devices be programmable so that they could be customized for the user's application.

The current product line features a 4-1/2 digit digital multimeter (DMM), a function generator, a voltage calibrator/reference, and a 20 MHz, 16-channel logic analyzer. This review covers the DMM, but the operation and functioning of the other devices are very similar. In all cases, what you get with the MetraByte solution is a \$700 to \$1,300 board that compares to an equivalent \$1,300 to \$3,200 IEEE-488/GPIB alternative. Obviously, the dollar savings is considerable. What you give up are slots in your PC because, unlike the normal IEEE-488/GPIB solution, you cannot daisy chain connectors and use only one interface board for many GPIB devices. (MetraByte offers such a board that will service up to 15 such devices, but that is another subject.)

MetraByte's PCIP-DMM

The digital multimeter from MetraByte is a full-length board that should fit in any PC, XT, or AT 8-bit slot. You must set one dip switch to specify the board address. The inputs to the DMM are isolated up to 400 volts from the computer and line power, and the board is fuse protected. The conver-

sion rate is 2.5 readings per second.

The specifications are DC voltage: 200 mV, 2V, 20V, and 200V with ± 0.03 percent accuracy; AC voltage: 200 mV, 2V, 20V, and 200V with ± 0.2 percent accuracy at 60 Hz; resistance: 200 ohm, 2 Kohm, 20 Kohm, and 2 Mohm with ± 0.03 percent accuracy; and AC/DC current: 2 mA, 20 mA, and 200 mA with ± 0.3 percent accuracy. While these ranges may not be quite as wide as a typical hand-held VOM, they are characteristic for such devices and quite sufficient for most applications.

Operating in the simplest mode, the DMM is a pop-up that occupies one-third of the screen. This allows other PCIP devices to share the same screen or lets you perform some concurrent DOS application. You select the mode (AC or DC), the measurement type, and the input range either by the keyboard or a mouse (a real convenience). A pop-up help window is also available. Finally, a relative mode allows readings to show as the difference from some predefined offset, and a hold mode stops the continuous updating of the display.

Installation of the board is straightforward and is aided by a utility program that shows how to set the address dip switch. Another program modifies the Config.sys file to install the two system drivers used by the software that comes with the board.

One problem I encountered was the inconsistency in the overrange indication. For example, an open circuit on the resistance range displayed as +199.95 in the 200 ohm range and +.+.+.+ in the 20 Kohm range. Outside of that the system worked very well. The complete documentation includes two significant sections, one on maintenance and repair and one on programming.

A programmable DMM. What makes this board so versatile is its programmability. While the manual covers the device registers so that you could write your own routines to manipulate the board and gather data, you need not do so. MetraByte has included a language-independent interface with the PCIP-DMM. After performing a standard file open for device "\$DMM," Basic, C, Fortran, and Pascal programmers can write

commands to the device to set or read the range, function, current type, and hold mode; set or read the relative setting; show or hide the screen display; read the last calibration date; wait for a new reading; and lock or unlock data entry from the keyboard or mouse.

A simple program fragment in Basic would look like the following:

```
10 OPEN "$DMM" FOR OUTPUT
   AS #1
20 OPEN "$DMM" FOR INPUT AS
   #2
30 ON ERROR 1000
40 PRINT #1, "Set 20 V dc" 'set input
   range
50 PRINT #1, "Read" 'set to read
   mode
60 INPUT #2, Volts 'read in a value
.
.
.
1000 INPUT #2, ERRNO, A$, B$ 'get
   error number and messages
1010 PRINT ERRNO, A$, B$ 'display
   error information
```

Programming hints are given for Microsoft and Turbo C along with Microsoft and Turbo Pascal. You can find Basic, Fortran, Pascal, and C sample programs on the floppy diskette that comes with the unit. I ran the program fragment given above using Microsoft's Quick Basic and was pleased at how simple it was to write code for this device. The built-in error handling was another plus because it retrieves an error number, error message, and command line stub to greatly aid debugging.

Summing up

This solidly built digital multimeter performs well. It is backed by a one-year warranty. Considering the programmable features plus the excellent software that comes with the unit, \$699 is an attractive price. When you consider that you can combine this unit with the other PCIP devices from MetraByte, you realize that the collection of boards can turn your PC into a versatile laboratory instrument.

Contact MetraByte, 440 Myles Standish Blvd., Taunton, MA 02780, phone (508) 880-3000.

Reader Service 21

COMPUTER

ILS — powerful software for signal processing

Interactive Laboratory System Version 6.0 is a sophisticated integrated software system consisting of more than 80 interactive programs. ILS, although designed for signal processing, can be useful for many other engineering and scientific applications because it allows direct control over data acquisition and analysis. But to use the system, you need not write special programs.

The main functions of ILS appear in Figure 1.

You can apply ILS in the analysis of any type of time series data. Typical applications include those shown in Figure 2.

Hardware configurations

ILS runs on

- DEC computers: VAX/VMS 4.3, MicroVAX/MicroVMS, VAXmate/MS-DOS, and VAXstation II, and VAXstation II/GPX with MicroVMS Workstation. You will also need the VAX Fortran compiler if you plan to modify ILS code. ILS requires a graphics terminal. The following terminals are supported (including color support for color terminals): DEC VT 24X and 34X series, VAXstation II, VAXstation II/GPX, VAXstation 2000 and 3000, and Tektronix 41XX series.
- SUN-3, using Sun-Unix 3.2, Sun-View, and Pixwin, with 8 Mbytes.
- Apollo Series 3000 and 4000, using Aegis 9.5, with 8 Mbytes, Domain/Dialogue User Interface Design and Management System, and Graphics Metafile Resource.
- Masscomp MC-5000, using Real-Time Unix (RTU) V3.1, 22 Mbytes disk space, and terminals GA600 (Integral), GA800 (IGP), and GA1000 (Aurora).
- IBM PC, XT, AT, and 100 percent compatibles, with 640 Kbytes of memory, PC-DOS 2.1 or MS-DOS 2.11, one double-sided floppy disk drive, one hard disk drive with 6 Mbytes, an 8087 or 80287 math coprocessor, a serial communication (RS-232) port, and one of the following graphics cards: IBM Color Graphics Adapter, IBM Enhanced Graphics Adapter, or Hercules Graphics Card.

The distribution media is 9-track tape and TK-50 cartridges. RX-50 distributions are also available. Documentation includes the following items and manuals: keypad layout card, menu tree, release notes, software product description,

introduction to ILS, installation guide, user's guide, programmer's guide, command reference guide, subroutine reference guide, and application notes on digital filtering, speech processing, and pattern classification.

The manuals are illustrated, well-organized, and easy to use. But, because ILS covers such a large number of fields from engineering and scientific disciplines, it is not possible to find all the details in the documentation. I also found a few minor errors in the manuals. Customer support includes on-line documentation, a telephone hot line for problem

solving, and periodic software updates and enhancements.

The installation of the system is simple, taking between 30 minutes and 2 hours. The price range is \$995 to \$20,000 (from PC to VAX computers). Prices for both Sun and Apollo systems range from \$5,100 to \$12,500, with extra node pricing at \$1,500.

Using ILS

ILS has two modes: menu and command. In the menu mode you can access

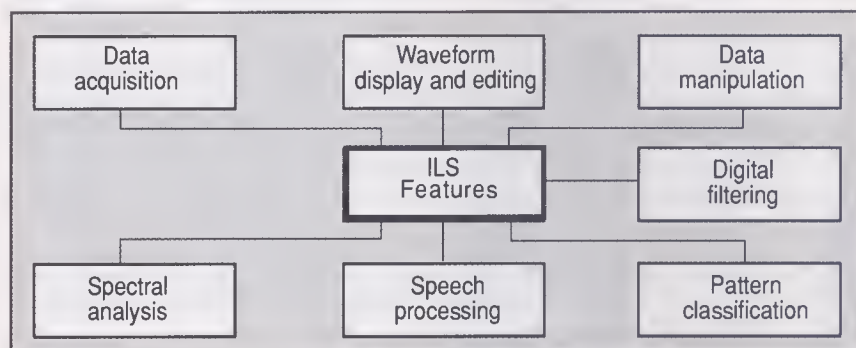


Figure 1. Main functions of Interactive Laboratory System.

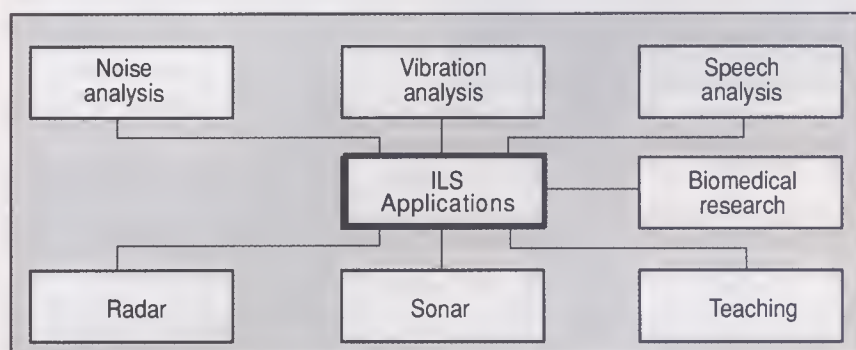


Figure 2. Typical applications of Interactive Laboratory System.

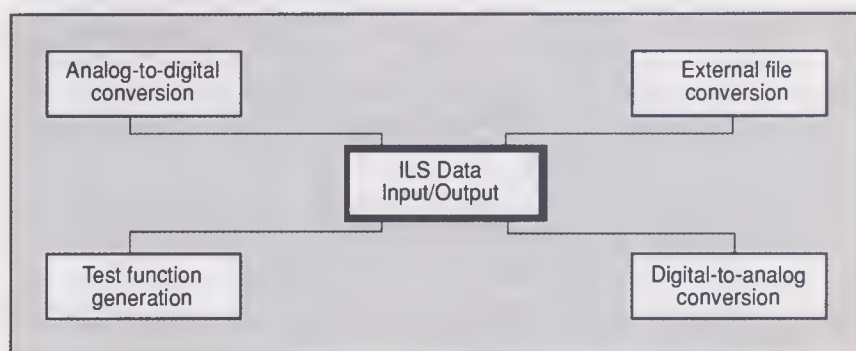


Figure 3. Choices for ILS data I/O.

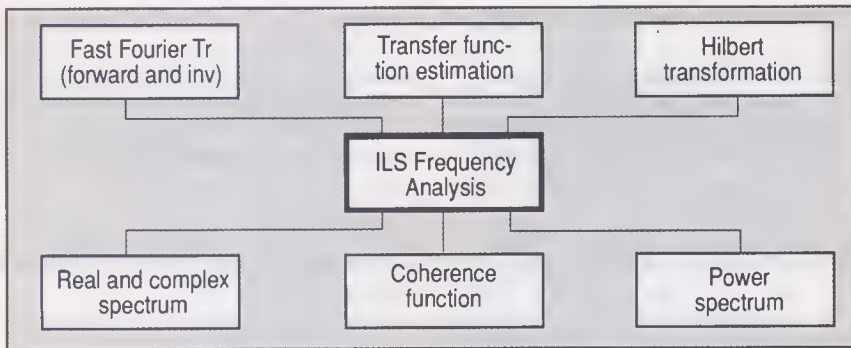


Figure 4. Frequency analysis techniques available in ILS.

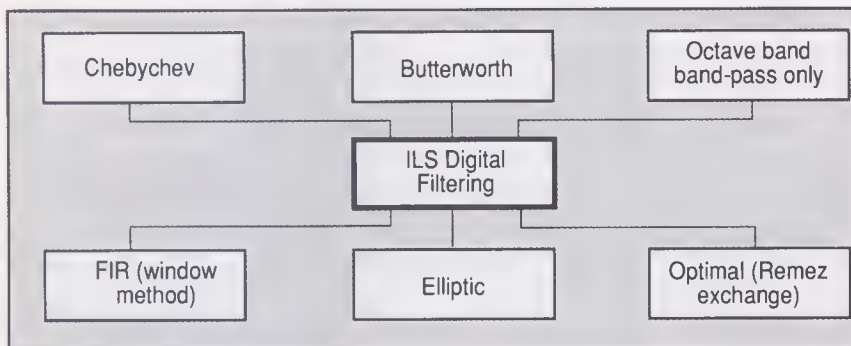


Figure 5. Digital filter types available in ILS.

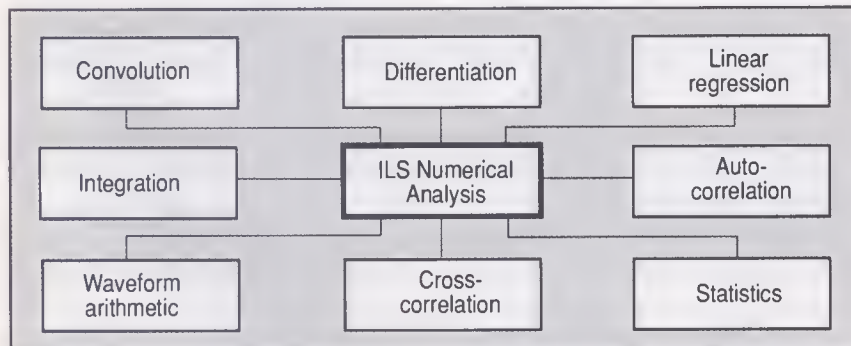


Figure 6. Mathematical operations in ILS numerical analysis.

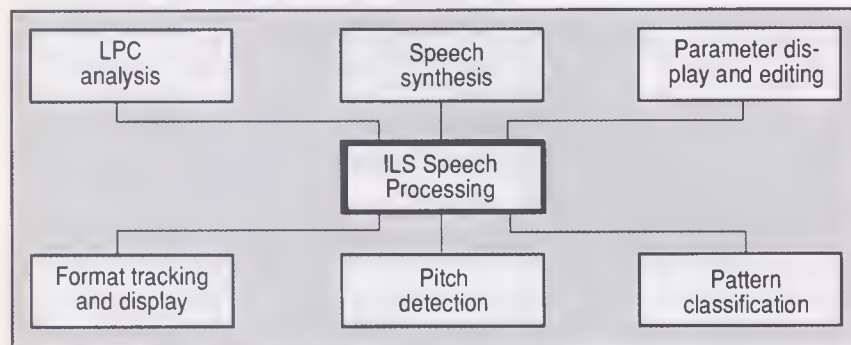


Figure 7. ILS algorithms available for speech processing.

the usual digital signal processing (DSP) functions. On-line help is available at every menu level, but the level of detail given is not always deep enough. The keypad control gives an easy menu traversal. You can also call menus directly, without multilevel passing.

The command mode is a bit more complex because you must learn a lot of the command syntax. The advantage of this mode is that a more experienced user can utilize the command concatenation feature to generate production runs for some applications.

If you have a VAXstation II, ILS will support its multiple graphics windows, and you'll be able to control the creation, deletion, positioning, sizing, and use of the windows. You can view and compare results from different analysis steps simultaneously. If the VAXstation is a GPX model, you can also control the display, text, grid, and background colors. For data acquisition and analysis, ILS supports DEC's real-time workstation, VAXlab.

Main capabilities

The data input/output section gives the choices shown in Figure 3.

The frequency analysis techniques available appear in Figure 4.

You can design and use filters — low pass, high pass, band pass, and band reject — of the types shown in Figure 5.

The numerical analysis section contains the mathematical operations shown in Figure 6.

The waveform displays available are time series display, zoom display, overlay display, 3D spectral display, and frequency display.

For data editing and transformation, the ILS has the following features: extraction with cursor, modification and scaling, wrapping or unwrapping phase, applying time delay, rectangular and polar conversion.

For speech processing there exist several powerful algorithms, shown in Figure 7.

One feature I really appreciated was source code availability. The ILS Fortran source code is provided, allowing you to modify or append the system. You can arrange to call ILS routines into your programs, or integrate your subroutines into ILS. Thus, ILS can be useful to many engineers, scientists, and computer professionals not only from the signal processing area, but also from other areas that use some of the advanced tools presented above.

ILS includes tools for converting external data files (in binary or ASCII) into ILS-compatible format. Support is also

included for converting ILS data files into external formatted files.

Summing up

ILS has been used in a number of widely different applications, including

- the development of a Cochlear Implant System for the profoundly deaf;
- the analysis of the tendency of the wings of an aircraft to vibrate under different aerodynamic and flight conditions;

- voice recognition of whales, porpoises, and birds as well as bat sonar and vocalization;
- performance quality standards for new communication technology, such as speech coders, synthesizers, compressors, switching equipment, modems, channel banks, and statistical multiplexers;
- speech recognition for telephone applications;
- testing models of buildings and offshore structures for wind load as well as wave load; and

- assistance for major reconstructive surgery to correct congenital cleft palates, an affliction that seriously affects speech.

I liked and can recommend Interactive Laboratory System to those involved in signal processing and those interested in advanced scientific and engineering applications.

Contact Signal Technology, 5951 Encina Rd., Goleta, CA 93117, phone (800) 235-5787 or (805) 683-3771.

—Michael Dediu

Reader Service 22

Solid-state, portable recorder

In my biomedical work I often need a portable recorder for gathering pressure signals and electromyograms (EMG). What I want is something small and convenient to replace a bulky chart recorder. Thus, when I received a demo disk from World Precision Instruments showing off its digital player and recorder called the DataSpan20, I knew I had to try one out.

The DataSpan20, shown in Figure 8, is about 5 × 7 × 2 inches — not quite small enough to fit in your pocket, but a reasonable size to move around from experiment to experiment. In many ways it resembles an oscilloscope, from its input gain and trigger controls to its display and sample rates. The input and output signals (as well as the trigger input/output) are made through the standard BNC connectors. Thus, I felt very comfortable using the device right out of the box without having to read the accompanying instruction manual.

How the device functions

The DataSpan20 consists of a micro-computer, an 8-bit A/D and D/A converter, a 9-pin RS-232 serial port, and up to 512 Kbytes of memory (the unit I reviewed included 64 Kbytes). During use, the device digitizes the input and saves it in its memory. At the same time, it converts the signal back to analog for simultaneous display on some external device (such as a scope or chart recorder). Obviously, the only difference between recording and playback is that the digitizer is not used in playback.

Because you can see the signal that you are storing in memory as you store it, you have the option of deciding whether to keep the current sample or to continue and try for another one. In fact, the storage modes assist in this operation because the "Single" mode simply records (or plays) until the end of memory

is reached. Thus, you can start (and stop) the recording whenever you wish and fill memory accordingly. Alternatively, in "Continuous" mode, the device repeatedly records (or plays), overwriting memory (or starting again in the case of play) until you push the stop button.

Sample rates range from one sample per second to as high as 50,000, and display rates are the same. But, since this is digitized data, the two rates need not be the same, allowing you to play back the sample faster, at the same speed, or slower than the sampling rate. Also, if you need it, the trigger can be either internal or external (TTL levels) and a delay of 0, 100, 500 or 1000 milliseconds is push-button selectable. Finally, selectable input gains of 0.5, 1.0, 5.0, and 10.0 give an input voltage range of ± 4 mV to ± 10 V.

Using the DataSpan20

I tested the DataSpan20 as a part of a number of different experiments I was running. The recorder worked flawlessly and was a big convenience over an oscilloscope or chart recorder. Particularly noteworthy was the selection of different sample and display rates because I could compress a slowly varying sample into a short trace, allowing me to determine the overall characteristics of my sample before looking at it in greater detail.

One thing missing in the first release I received was software to display the acquired data on my PC. Fortunately, just in time for this review, WPI released not one but two such packages: DataView for the PC and MacSpan for the Macintosh. Each is uniquely tailored to the features of the machine, but each offers the essential features that make the DataSpan20 connected to a personal computer a very versatile instrument.

Both software packages transfer the

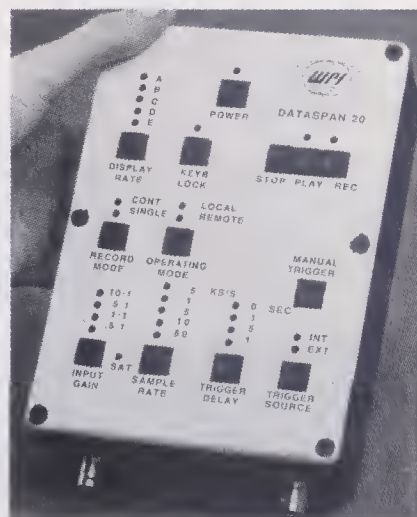


Figure 8. World Precision Instruments' DataSpan20 portable recorder.

digital data to the PC using the serial port. MacSpan is tailored to the graphics interface of the Mac and offers scrolling, zoom in and out, scaling, continuous display of time and magnitude, and relative measurements of the waveform displayed. Of course, other standard Mac features are supported, like the clipboard with cut and paste and a notebook to annotate the signal displayed. A print command provides hard copy of the trace if desired. By turning on the overlay feature, you can acquire a signal from the DataSpan20 and simultaneously display it and the currently selected waveform.

DataView offers similar features for transferring the waveform to the PC, as well as scroll, zoom, scale, time and magnitude display, relative measurements, and print. To use the display mode, you will need a Hercules display adapter and monochrome monitor.

One significant difference between the two software packages is that the PC version can show up to four previously transferred traces simultaneously, each with its own scaling and zoom parameters. I found this feature quite impressive and thus preferred DataView to MacSpan.

The standard software that comes with DataSpan20 is called Dterm. It provides for the remote setting of all the device's capabilities, sample rate, trigger, record or play, and so forth, and will also transfer files back and forth between the PC and the recorder. These software features are built into DataView and MacSpan as well. The base system price for the DataSpan20 is \$975. DataView and MacSpan each cost an additional \$50. You get not only the versatile software, but also the serial interface cable to connect your recorder to the PC or Mac.

More details

I mentioned a serial port on the DataSpan20 and I also described the software that plots multiple samples. How is this done? The answer is that the serial port is used to move the digitized signal between the recorder and your PC or Mac (I'll use PC in the rest of this article, but all things apply equally well to the Mac). Thus, you can capture one signal, store it on your computer, take another, store it, and so on. You can, of course, download a sample to the DataSpan as well. But what do you do if the experiment is in one room and the computer in another? The designers of this device solved that problem nicely by using static RAM and a nicad battery backup on the system board.

Another interesting and thoughtful design feature is that you can "program" the device from your computer. As I mentioned earlier, the DataSpan20 uses the serial port to set up the rates and modes of the device using the PC's display and the keyboard. By the way, the serial connection to the PC can be made at all the standard rates, from 100 to 38.4 Kbaud. I used 19.2 Kbaud (the suggested rate), as well as 38.4 Kbaud, without a hitch.

The manual that comes with the DataSpan20 gives most of the details and specifications of the device. In the first release, the instruction manual included two appendixes that explained the file format and the operational mode. This means that you could generate a sample and download it into the DataSpan20 to be carried to a remote location for later playback. It also means that you could write a program to completely automate the process of setting up the recorder and dumping the data file. While the infor-

mation is still available (by request), WPI found that most users didn't need it and dropped it from the instruction manual. However, they did include a conversion command in DataView so that the internal format can be made into standard ASCII files and read by Lotus 1-2-3 or Asyst.

Summing up

The DataSpan20 is easy to use, performs as advertised, and is very practical. During testing, several colleagues became interested and expressed a desire to try it out within their research environment. I asked them about the price, and they agreed with me that the device is reasonably priced and comparable in features and flexibility to a small oscilloscope.

I did run into some minor problems worth reporting. First, the use of the terms "input" and "output" confused me. Sometimes it seemed that the manual referred to the PC, while other times it seemed like it referred to the DataSpan.

Second, I found inconsistencies between what the manual claimed and what you could really do. The manual and software would benefit from a bit more testing.

Third, on the Mac, while scrolling a waveform I would occasionally get a double trace because the old trace was not cleared completely. Also, the full view of the waveform showed considerable aliasing, so what I saw bore little resemblance to the actual signal.

Finally, the Hercules emulation mode of my Paradise VGA Professional display adapter worked at about the 95 percent level. This meant that, while I could display, scroll, zoom, and mark the traces, I couldn't read the relative voltage differences, some keys became inoperative (such as the one to scroll backwards two screens), and often it took repeated keystrokes to exit a submenu. WPI assured me that these problems arose from a compatibility problem, which I don't doubt. This means you will need to invest in a Hercules display adapter if you want to use the DataSpan20 with a PC, since currently the device does not support any other display mode.

Even though I did find several items to criticize, I recommend the DataSpan20 as a valuable laboratory instrument with a wide range of possible uses. This recorder deserves serious consideration as an alternative to other boards and devices that make your personal computer into a digital recorder and waveform display/analysis device. In addition, the company offers an interesting range of devices from pressure, temperature,

force, and displacement transducers to an analog transducer amplifier to a number of other speciality devices. At the very least, if you use this type of equipment, you should request the WPI data processing and signal conditioning catalog.

Contact World Precision Instruments, 375 Quinnipiac Ave., New Haven, CT 06513, phone (203) 469-8281.

Reader Service 23

Review notes

Emulation. I tested and liked the poly-Star/320 (Version 1.0) and 240 (version 2.0) terminal emulation programs for using a PC with any DEC mainframe/minicomputer that requires DEC terminal emulation. poly-Star includes both a script language and special remote-control sequences to initiate file transfers using Polygon's own proprietary poly-XFR protocol or the Kermit protocol. For pure text files, poly-Star also offers streaming file transfer, which does not require a special host program. Star also offers multiple sessions, both network and asynchronous, which you can use to switch between different hosts or between different applications on the same host. System requirements are a PC, XT, AT, or PS/2 compatible computer, DOS version 2.0 or greater, 512 Kbytes of RAM (640 Kbytes of RAM recommended), and a graphics display. The price is \$299 from Polygon, 1024 Executive Pkwy., St. Louis, MO 63141, phone (314) 576-7709. — *M. Dediu*

Windows Express. In my review of Windows Express (*Computer*, August 1988, p. 85), I recommended this utility very highly. The latest release from hDC Computer, 15379 NE 90th St., Redmond, WA 98052, phone (206) 885-5550, takes this excellently done software to new heights. Key new features include lower memory requirements, an icon library, more options to specify the contents of screen displays, on-line help, an expanded and more colorful graphics interface, automatic saving of preferences (such as window sizes) between sessions, and an expanded set of manuals. An upgrade costs \$20 plus shipping and handling. To my mind there isn't much point in running Microsoft Windows without installing this package. — *R. Eckhouse*

Connectivity card. Another way to communicate between a PC and other computers is with a \$395 Multi Term Connectivity (MTC) card along with the STS-200 communications software available from StarSoft Technologies, 14884 Berry Way, San Jose, CA 95124, phone

(408) 727-7074. This unusual board supplements either a monochrome or color graphics adapter card. It is designed to use IRQ 2 on the PC, so conflicts might occur when other cards (such as a clock/calendar board or a multifunction card) are installed. The combination of hardware and software provides high-resolution text display with features that software alone cannot supply, such as: 80 or 132 displayable columns, 27 or 44 lines of text, smooth scrolling, all standard screen attributes, bold and underline, double high and double wide characters, and selectable cursor. The MTC board plus software generates the essential video display characteristics of the Hewlett-Packard 2392/2394, DEC VT 220/320, Wyse 50, Qume 101, Televideo 925/950, and others. There are programmable function keys, communication protocols, file transfer (using Kermit or XModem), split screen for dual host communication (possible with two serial ports installed), and a screen server. The board communicates using the serial communication ports already installed on the PC. — *M. Dediu*

Emulator for the IBM PC. Looking for a Tektronix 4105 and DEC VT220 emulator for the IBM PC? EM4105 version 3.52-1.48 from Diversified Computer Systems, 3775 Iris Ave., Suite 1B, Boulder, CO 80301, phone (800) 247-3834 or (303) 447-9251, might be what you want. In VT102 mode, the emulator offers scrolling regions, video attributes, double width characters, local printer support, character insert/delete, and full keyboard emulation. In VT220 mode, it emulates the expanded VT220 keyboard, user-defined function keys, compose key, multinational character sets, and 8-bit control sequences. In TEK mode, you get the graphics functions of a 4105 graphics terminal supporting 8 colors from a palette of 64, polygon fill, line styles, independent dialog area, markers, graphics input mode, and 4010 compatibility. I liked the additional features of split-screen horizontal scrolling, screen scroll back, DOS hot key, softkeys, automated modem dialer, phone directory, data logging and replay, file transfer, command files, pan/zoom (available in version 1.4), graphics picture files, VT640 emulation, 16 simultaneous colors, loadable graphics drivers, and high-resolution hard copy. The documentation is detailed, but could be a little more accessible. The price is \$349. — *M. Dediu*

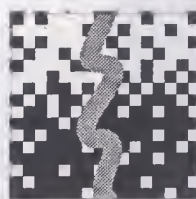
Tax return. If you prepare your own tax return, you should probably do it on a computer. Unless you have a very simple return, a tax program will save time and reduce errors. But which program should

you use? I have tried several tax preparation programs and would make the following recommendations. If you have an Apple IIGS, a Macintosh, or an IBM compatible capable of running Microsoft Windows (hard disk, graphic display, mouse, and AT speed), get TaxView (or MacInTax as it's called for the Mac) from SoftView, 4820 Adohr Lane, Suite F, Camarillo, CA 93010, phone (408) 388-2626. Exact replicas of the tax forms are displayed on the screen for you to fill in as you would a paper form. All arithmetic is done dynamically as you enter values, and all of your forms are automatically linked.

For those with a smaller, character-oriented PC compatible, you won't be able to use TaxView, so get TurboTax from ChipSoft, 5045 Shoreham Pl., Suite 100,

San Diego, CA 92122, phone (619) 453-8722. It lacks exact form replicas, but they're similar. Information need only be entered once because everything is linked. TurboTax calculations are not automatic, but are made when you give a calculate command.

Both programs print IRS-acceptable replica forms (slowly), and both warn you if any information is missing. They also feature on-line operating and tax help, and I would expect anyone familiar with tax forms to be able to use either program without the manual. Each program comes with about 40 forms and schedules, and you should check to be sure all those you need are included. Also, be sure there is a supplement for your state or you will have to prepare your state return manually. — *L. Press*



First International Symposium on Integrated Network Management

May 14-17, 1989 • Boston, Massachusetts

Improving Global Communication Through Network Management

Sponsored by the International Federation for Information Processing (IFIP)
Hosted by The MITRE Corporation and NIST (formerly NBS)

This symposium, a unique combination of high-level technical sessions and presentations by industry-leading vendors, features:

- Technical tracks on management functions, architectures, and models; standardization issues and user requirements; integrated management of multi-vendor networks; and artificial intelligence in network management
- In-depth tutorials on comparative analyses of network management products, ISO standards, management of manufacturing networks, integrated network security, and tools for performance and fault management of LANs and WANs
- Special presentations by Vinton Cerf, Vice President, Corporation for National Research Initiatives; Lt. Gen. John Myers, Director, Defense Communications Agency; John Gibbons, Director of the Office of Technology Assessment; and John Crecine, President, Georgia Institute of Technology

Corporate Patrons: Digital Communications Associates, Inc., MCI Telecommunications Corporation, Atlantic Research Corporation, Boeing Computer Services, Digital Equipment Corporation, OSI/NM Forum, Codex, Wang Laboratories, Inc., IBM, NYNEX Corporation, Data General Corporation, Avant-Garde Computing, Inc.

For advance program and registration information, contact Hershey Young,
National Institute of Standards and Technology
Building 225, Room B217, Gaithersburg, MD 20899
(301) 975-5267

NEW PRODUCTS

Contact or send releases to Nancy Hays, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, Compmail+, n.hays

Apollo combines RISC and graphics in Series 10000VS

Apollo Computer has announced the Series 10000 Visualization System based on its Parallel Reduced Instruction Set Multiprocessing, or PRISM, architecture.

The Series 10000VS contains from one to four processors, each reputedly able to achieve 1.5 million 3D graphic transforms per second. The CPUs are integrated with a RISC drawing engine, which executes — without microcode — drawing tasks common to all applications. The graphics engine is coupled to the CPUs through a 64-bit wide, 150 Mbyte per second bus.

The central processor incorporates an

integer processor, ALU and multiplier floating-point processors, a 32×64 8-port floating-point register file, a 128-Kbyte 64-bit-wide instruction cache, and a 64-Kbyte 64-bit-wide data cache. Memory consists of 8 to 128 Mbytes of RAM and is 32-way interleaved.

Bus include an 8-slot, 64-bit-wide, 150-Mbytes/s X bus; a 6-slot, 32-bit-wide, 12-Mbytes/s VMEbus; a 4-slot, 16-bit-wide, 3-Mbytes/s PC AT bus; and a dedicated diagnostic bus.

The Series 10000VS features programmable 40- or 80-bit plane con-

figurations, programmable Z-buffer depth up to 32 bits, subpixel addressing, quadratic shading that applies second-order functions to color interpolation, texture mapping, hardware support for dithering, a $1,280 \times 1,024$ 70-Hz color monitor, and configurations with up to four striped 348- or 700-Mbyte ESDI disk drives.

Prices for a single-processor Series 10000VS start at \$94,900 for a 40-plane system and \$104,900 for an 80-plane system. Shipments are scheduled for the second quarter of 1989.

Reader Service 30

DEC upgrades with VAX 6300 Systems

Digital Equipment has enhanced the VAX 6200 line with its introduction of the VAX 6300 Systems, the company announced. The new systems reportedly deliver up to 35 percent higher performance and have twice the expansion range of the VAX 6200 computers.

The new systems include a 14-slot, 100 Mbytes/s system interconnect, which provides a mix of up to six CPUs, up to 256 Mbytes of memory, and up to six VAXBI adapters. According to the company, processor and memory upgrade packages permit upgrading from the entry-level VAX 6310 to the top-of-the-line VAX 6360.

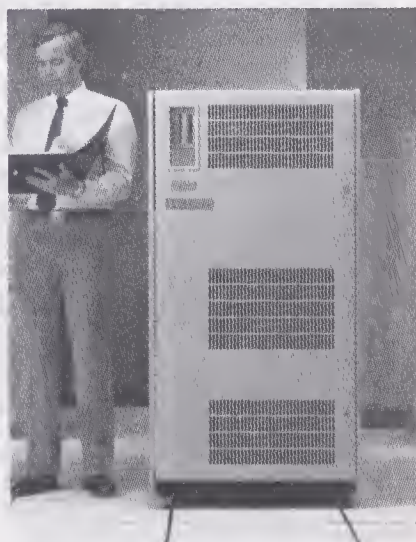
The new systems also feature up to 256 Mbytes of shared ECC memory.

The VAX 6300 Systems support the VMS operating system on all models. Ultrix-32 is available on the VAX 6310 and 6320 models.

Prices start at \$155,200 for the VAX 6310 Ultrix-32 base system with one CPU, two VAXBI channels, one TK70 cartridge-tape load device, 32 Mbytes of ECC shared memory, a DEBNA Ethernet controller, a KDB50 4-port disk controller, an Ultrix-32 64-user operat-

ing system license, and a DECnet/Ultrix license.

Reader Service 31



Digital's VAX 6300 systems join the midrange of DEC computers as the second generation of the VAX 6000 series.

Starter kit for CADD

Generic Software has combined four products into the CADD Starter Kit for first-time users of computer-aided design and drafting software. The kit contains Generic CADD Level 2, a workbook tutorial, Generic's Basic Home Design symbol library, and a disk of tutorial and sample drawings.

System requirements are 384 Kbytes of RAM, DOS 2.0 or higher, one floppy disk drive, and a graphics card (Hercules, ATT 6300, CGA, EGA, or VGA).

The starter kit costs \$159.95.

Generic has also announced Estimator, a software package that assists in job estimations and bid preparations. It works with CADD Levels 1, 2, or 3, or can be used alone. It reportedly can estimate up to 16,000 different items and generate up to 20 reports.

Generic Estimator, developed by the company and Contractors Management Systems, comes with a manual, tutorial, sample bids, and on-line help. It costs \$295.

Starter kit: **Reader Service 32**
Estimator: **Reader Service 33**

API kit delivers X.400

Prime Computer has announced the X.400 Application Programming Interface Development Kit, which allows 50 Series superminicomputer software programmers to develop electronic mail, gateway, and other messaging systems within local and wide area networks. The product is based on the 1984 X.400 international communications standard.

According to the company, the kit's data communication protocols and application programming interface permit programmers to build messaging systems that store, deliver, and exchange information among different vendors' systems. Information can be formatted as text, graphics, digitized voice, image, and EDI transactions.

The X.400 API system operates across the X.25 protocol and under Prime's Primos operating system. Future products are planned to support Prime's Unix-based systems and Ethernet LANs.

The API kit costs \$350,000 for a complete software license, or \$150,000 for an initial development license, with subsequent sublicense fees from \$1,000 to \$4,000 according to the configuration. Software service costs \$375 per month for all licenses. Service includes telephone support, updates, and on-site remedial and installation support.

Reader Service 34

Unisys SCAMP moves A Series to desktop

Unisys has housed the Single Chip A-Series Mainframe Processor, or SCAMP, in a Personal Workstation² Series 800 personal computer to produce the Micro A, which the company calls a desktop mainframe. The SCAMP contains the 48-bit A Series processor with 2.5 Mbytes of static RAM. The chip occupies an AT-size expansion board, as do 12 Mbytes of A Series memory and the I/O control logic.

The SCAMP uses the standard A Series Master Control Program/Advance System, or MCP/AS, operating system. The PW² configuration relies on a 16-MHz Intel 80386 processor and the Microsoft OS/2 operating system. It serves as the I/O processor and maintenance subsystem for the Micro A. A separate data communications host adapter board handles data communications.

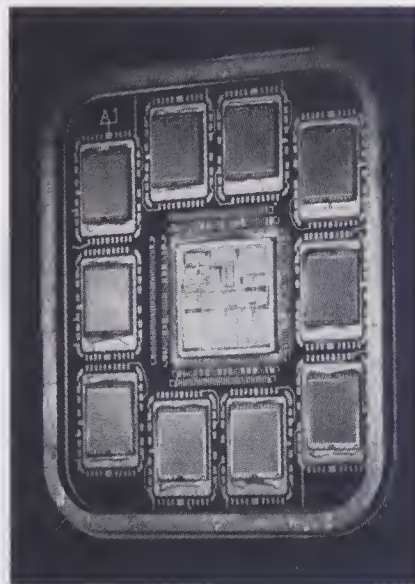
The Micro A includes 3 Mbytes of main memory, a 20-Mbyte hard disk drive, and a 1.2-Mbyte floppy disk drive, plus a 280-Mbyte SCSI disk drive and a 150-Mbyte SCSI 1/4-inch cartridge tape drive. It also features an EGA board.

The SCSI board supports a maximum of seven SCSI devices, which means five additions to the base system. The DCHA board is an 80286-based coprocessor board with 512 Kbytes of dedicated memory. It supports up to

four separate lines through RS-232 connectors and can run a maximum of three communications protocols concurrently.

The Micro A costs \$20,365 for CPU, keyboard, and monochrome monitor. An MCP/AS license costs \$5,000 and includes OS/2 software.

Reader Service 35



The Unisys SCAMP package contains the Micro A mainframe instruction processor and 10 256-Kbit SRAM chips.

Wang superminis handle high-volume computing

Wang Laboratories claims that it has optimized its new superminicomputers, the VS 10000 Series, for high-volume computing by incorporating a new disk subsystem and system management software. The new models — 50, 75, and 100 — are reportedly based on emitter-coupled logic technology with custom gate arrays, VLSI microprocessors, and the mainframe VS instruction set.

The VS 10000 systems use a 90-MHz clock rate and an I/O bus capacity of 30.3 Mbytes per second. The arithmetic unit is both a floating-point accelerator and a decimal-arithmetic accelerator.

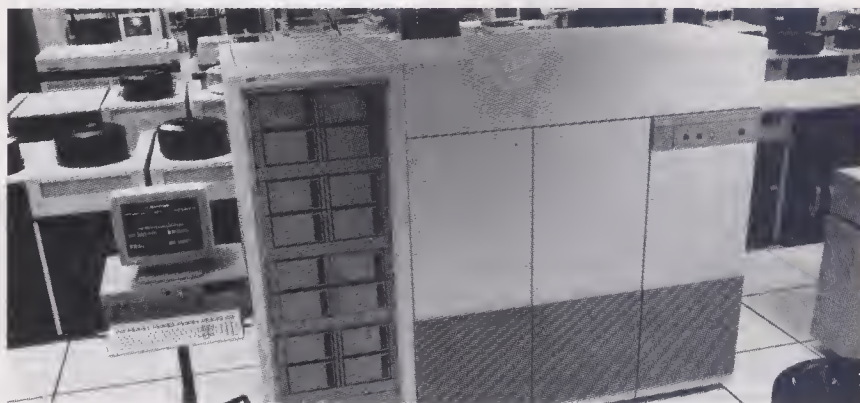
Other features include 32 Kbytes of write-back cache memory in the CPU, up to 64 Mbytes of addressable memory with physical accommodations for up to 256 Mbytes, a 128-bit-wide memory bus that supports 128-bit read and 64-bit write operations, an independent 80286-based support control unit, and

up to 15 intelligent I/O controllers.

Prices start at \$395,000 for the Model 50 with 16 Mbytes of memory. The Model 100 with 32 Mbytes of memory costs \$670,000. Licenses for the VS

operating system cost \$15,500 for the Model 50, \$23,250 for the Model 75, and \$31,000 for the Model 100.

Reader Service 36



Wang Laboratories' VS 10000 Series superminicomputers (shown here with the VS Mass Storage Subsystem) target large-scale data processing and integrated image processing.

New Mac supports new expansion options

Apple Computer has announced the Macintosh SE/30 personal computer based on the 16-MHz Motorola 68030 microprocessor and 68882 floating-point coprocessor. According to the company, the SE/30 is compatible with MS-DOS and OS/2 and provides RAM expandability up to 8 Mbytes.

The Macintosh SE/30 includes Apple's FDHD (Floppy Drive High Density), an internal 1.4-Mbyte floppy disk drive that reportedly reads, writes, and formats MS-DOS, OS/2, Macintosh, and Apple II ProDOS disks.

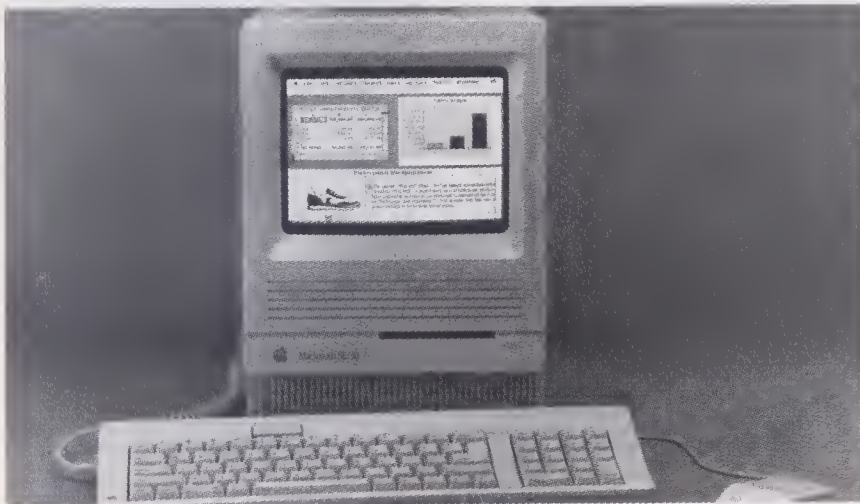
The SE/30 expansion slot architecture supports expansion options not previously available on the Macintosh, according to Apple, such as video,

memory parity, high-speed communications, and sound. The motherboard contains a sound chip.

The company offers three configurations of the Macintosh SE/30. The first option has 1 Mbyte of RAM and a 1.4-Mbyte FDHD for \$4,369. The second option has 1 Mbyte of RAM, FDHD, and an internal 40-Mbyte hard disk for \$4,869. The third option has 4 Mbytes of RAM, FDHD, and an 80-Mbyte hard disk for \$6,569.

The new PCs come with Macintosh System Software 6.0.3, Multifinder, and Hypercard.

Reader Service 37



Apple claims that its new Macintosh SE/30 is up to four times faster than the SE.

Ada Software Repository out on floppy disk

Advanced Software Technology is distributing the Ada Software Repository on MS-DOS-format high-density floppy disks. The ASR is a repository of Ada programs, software components, and educational material established on the Simtel20 host computer on the Defense Data Network.

According to the company, the ASR serves two basic roles: to promote the exchange and use of Ada programs and tools and to promote Ada education. The ASR includes more than 60 Mbytes of Ada source code and information.

The public domain software included in the ASR was formerly available only over DDN. Advanced Software Technology catalogued the library onto 19 disks in a compressed format. Users will need a hard disk to uncompress the files.

The disks cost \$10 each for shipping and handling.

Items available are Data Base Management, Math Library, Compilation Order Tool, Benchmark, Software Components, Forms Generator, Software Metrics, Pretty Printers, Pointers, Ada Language Reference Manual, Educational, APSE Interface Set, Artificial Intelligence, Cross Reference Tool, Program Design Language, Communications Tools, Menu Generation, Graphical Kernel System, Virtual Terminal, Management Tools, Ada-SQL, Simulations, Debuggers, Style Checkers, Editors, Spelling Checkers, and ASR Newsletters.

Reader Service 38

System speeds design review

Microsystems Engineering offers TechView, a CAD/CAM viewing and annotation system, for industrial applications. According to the company, TechView provides an electronic means of viewing and annotating industrial CAD drawings.

Reviewers can create annotation files and transmit them as CAD overlay files to the original designer. The system uses a copy of the drawing, leaving the CAD database intact.

TechView also permits local printing or plotting of the CAD drawings. It supports Postscript printers.

The system's graphic processor supports IGES, CGM, HPGL, PICT, and Tektronix.

Features include a collection of annotation symbols, plus predefined and user-defined annotation symbol libraries, and the ability to pan, zoom, and scale annotations.

TechView runs on IBM PC-XT, AT, or compatible computers and the DEC VAXstation. A Macintosh version is scheduled for the second quarter of 1989. The system uses a mouse.

TechView costs \$1,395.

Reader Service 39

ELXSI enters midrange

ELXSI has announced the System 6000 modular midrange computers. The five models are hardware and software compatible with the company's System 6400 computer systems.

The System 6000 includes single or dual CPUs, from 16 to 512 Mbytes of main memory, and system components that plug in to the company's 320-Mbytes/s system bus.

Performance reputedly ranges from 12 Whetstone MIPS for a system with a single 6420 CPU to 80 Whetstone MIPS for a system with two 6460 CPUs.

Model S6020-1, scheduled for availability in March, costs \$295,000 and supposedly achieves 12 Whetstone MIPS performance. Model S6020-2, available in May, reaches 24 Whetstone MIPS and costs \$395,000. Models S6060-1 (40 WMIPS, \$445,000), S6062 (52 WMIPS, \$545,000), and S6060-2 (80 WMIPS, \$695,000) will be available in October, according to the company.

A basic System 6000 comes with one or two CPUs, 16 Mbytes of main memory, an I/O processor, an 823-Mbyte disk drive, a 75-ips tape unit, and an asynchronous terminal controller.

Reader Service 40

CimStation upgraded

Silma has upgraded CimStation with release 4.0. The software aids in the design, programming, and simulation of automated work cells. According to the company, manufacturing engineers can use simulation models of industrial robots and equipment to simulate the operation of an automated work cell. They can then generate equipment programs for the robots and other equipment in the work cell.

Features added to release 4.0 include the ability to deform the surface of 3D models of parts to simulate product fabrication and machining, automatic collision avoidance, task-oriented programming, and user-programmable application menus.

CimStation 4.0 runs on workstations from Apollo Computer, Control Data, Intergraph, Prime Computer, Sun Microsystems, and Silicon Graphics.

Reader Service 41

Windows CASE tool controls object-oriented repository

Meta Systems has announced QuickSpec, a PC/Windows CASE tool for accessing and controlling an object-oriented repository. Using QuickSpec, systems analysts and designers can reportedly use PCs to enter, edit, and review project information and specifications in an object-oriented CASE repository.

According to the company, the repository can maintain information from different methodologies and diagram types, from all phases of the systems development life cycle. The repository supposedly uses a representational model broader than that proposed under IRDS.

Several applications can run simultaneously with QuickSpec. This permits cutting and pasting between the program and other applications, as well as viewing and editing through multiple windows.

QuickSpec runs under operating environments supported by Microsoft Windows, such as an IBM PC-AT using an 80286 or 80386 processor or IBM PS/2 models 50, 60, 70, or 80 with 640 Kbytes of memory, an appropriate monitor, a hard disk drive, and a mouse.

QuickSpec costs \$3,500.

Reader Service 42

Vision system handles industrial inspection

QC Vision from Image Inspection reportedly provides the image-processing hardware and vision software needed to inspect objects of any size.

The system consists of a camera; dual monitors to display images, instructions to operators, inspections, and inspection statistics; vision and image-processing software; a keypad for the production operator and a keyboard for development work; and serial I/O, printer, and floppy-disk ports.

The system is housed in a filtered-air-cooled enclosure.

According to the company, inspection

speed depends on the application, but a typical rate is two items per second. QC Vision provides continuous analysis of inspection statistics and user-selected statistics.

Users can choose to have alarms activated if preset reject levels are exceeded. Users can call up a different inspection routine for each product.

A screen display provides on-line help functions.

The system also provides development facilities for designing and testing inspection routines.

Reader Service 43

Vision software targets surface-mount market

Cognex offers vision software targeted to the surface-mount market in its SMD Tools, which work with a Cognex machine vision system to guide placement of surface-mount devices onto the circuit board.

According to the company, the SMD Tools use advanced gray-scale image processing techniques. The tools work in conjunction with any Cognex single-board vision processor, employing edge-finding algorithms to locate each lead on the surface-mount device before determining device position. The company claims that this approach permits fast and accurate placement even for devices with slightly bent leads.

The software consists of several modules that can be called from within a developer's C language application program. The developer can set new module parameters, replace modules, or add new modules.

Set-up modules allow users to provide the vision system with descriptions of the components and simplify calibration

of the camera or optical system. A device description module allows the developer to describe each component in geometric measurements rather than requiring the user to show each device to the system. The calibration function reportedly compensates for camera misalignment, as well as for optical distortions at the periphery of the field of view.

The image acquisition module allows the system to view the part in a single camera field or to direct a motion controller to move the part or camera to capture images of larger components in multiple fields of view. A part location module then determines the device's approximate position, while a lead location module determines the position of each lead.

The SMD Tools are scheduled for availability in mid-1989 for \$2,000. Developers have the option to purchase the SMD Tools software plus prerequisite software for \$4,000.

Reader Service 44

OS targets automation systems market

Siemens AG has an operating system designed for the automation systems market and based on the Unix System V standard, according to the company. The operating system, called Sorix, runs on the Siemens Sicomp SX 32-bit microcomputer system. It is not available in the US.

The system reportedly features the real-time capabilities of preemptive scheduling, memory-resident specification, quick semaphores and events, and high-speed external file access. It sup-

ports multiprocessing and high-level programming languages.

Sorix resides on EPROM or an external store.

Software is developed in a PC environment. The host for individual stages such as compiling, assembly, editing, linking, or programming of EPROMs is a Siemens Sicomp PC 16-20, PC 32-20, or other AT-compatible PC.

Reader Service 45

Compiler converts C source code to Postscript

Unipress Software has announced the C2PS compiler, which converts C source code into Postscript code. It reportedly allows programmers to write Postscript applications in C, thus facilitating the creation of windowing applications.

The company developed C2PS for use with Sun Microsystems' NeWS window system and other environments requiring Postscript code, such as Adobe Systems' Display Postscript, x11/NeWS, and Postscript printer applications.

Developers write NeWS programs in

two parts: the application (in C) and the graphic interface (in Postscript). With C2PS, programmers can write both parts in C and translate the graphic interface to Postscript.

Used with the Unipress Emacs editor, C2PS lets users edit and view C code in one window and view the produced Postscript code in another.

Binary code for C2PS costs \$2,995 and source code costs \$14,995. University prices for C2PS are \$995 for binary code and \$4,995 for source code.

Reader Service 46

CASE tool extends Ada to meet DoD 3405.2

Software Systems Design offers a CASE tool that reportedly extends the Ada language to satisfy DoD mandate number 3405.2. The set of 25 integrated tools is called Ada Design Language.

AdaDL provides tools to quantify the complexity of the overall software design and the executable Ada code, catch and flag design errors, compile high-level abstractions in structured English language and executable Ada code, comply with DoD mandate

3405.2, and produce an audit trail.

According to the company, AdaDL is a superset and extension of Ada. It adds a definition dictionary, high-level pseudo code, and project management capabilities, including documentation capabilities.

Prices range from \$7,150 for host computers like Sun and Apollo to \$15,700 for large systems.

Reader Service 47

Development system supports RISC processors

Hilevel Technology has announced development system support for the Motorola 88000 RISC processors. The DS5000-RISC Development System supports both the 88100 RISC processor and the 88200 cache/memory management units.

The system can trace activity on either or both the processor bus and the program memory bus. It supports trigger or breakpoint conditions for any combination of activity on the processor bus or memory bus or between the two.

The target interface and target control are provided in the C88000 probe. The probe connects to the target system through a controlled impedance flex circuit. The probe also supports zero-wait-state overlay memory to allow program development and debugging to begin before the target hardware is completely operational, according to the company.

Once the target memory system is operational, the DS5000-RISC allows downloading of programs directly into the target memory system for testing,

debugging, and performance analysis.

The DS5000-RISC supports Hilevel's Extended Acquisition, which allows indefinite performance analysis without interruption of target system operation. According to the company, Extended Acquisition captures trace data at up to 50 MHz and uploads it to a host computer using alternating 4-Kbyte buffers.

The development system also supports standard performance analysis capabilities. It supports full histogram of the target system's operation, measured execution time with a 15 ns or 250 ns resolution, and a memory usage map that can be applied to the instruction memory or the data memory.

The DS5000-RISC ranges in price from \$28,000 to \$40,000, depending on overlay memory selection, number of trace options, and other options. Each system supports up to 256 channels of logic analysis plus options. It runs on an IBM PC or Sun computer using MS-DOS.

Reader Service 48

CAD system targets 386 PCs

CADAM has announced Micro CADAM Plus, a CAD system designed for 80386-based PCs. According to the company, the software provides access to memory beyond the 640 Kbytes addressed by DOS. This reportedly allows users to create models as large as those possible with the company's mainframe-based CAD system, CADAM Interactive Design.

Micro CADAM Plus adds an overlay function that allows users to create and manipulate a design in separate but related sections, yet display and plot them as a unit. It also features advanced area fill capabilities.

The program reportedly accommodates major LANs. It also permits exchange of data with mainframe-based and Unix-based CADAM systems by means of the optional Data Transfer product. Moreover, designs and drawings produced on Micro CADAM Plus are upwardly compatible with those products.

Foreign alphabets such as Chinese, Japanese, and Hebrew are available for annotation. Users can also incorporate geometric tolerancing symbols in text notes and annotations.

Micro CADAM Plus runs on 80386-based PS/2 and Compaq models. System requirements include 640 Kbytes of memory and 4 Mbytes of extended memory, a math coprocessor, and a hard disk.

Micro CADAM Plus costs \$3,995. The software, supplied on floppy disks, comes with documentation, a self-study course, and a mouse.

Reader Service 49

CASE benefits evaluated

Index Technology's CASE Project Pack helps users evaluate the benefits of CASE technology on a pilot software development project. The program includes rental of the company's Excelsior systems analysis and design software along with consulting, training, and support.

Under CASE Project Pack, Index Technology takes organizations through four phases of CASE implementation: project planning review, pretraining preparation, Excelsior and structured analysis training, and pilot project support.

Prices range from \$20,000 to \$65,000 according to the size of the project and the time required to complete it.

Reader Service 50

IC Announcements

Company, Model, Function

Comments

R.S. No.

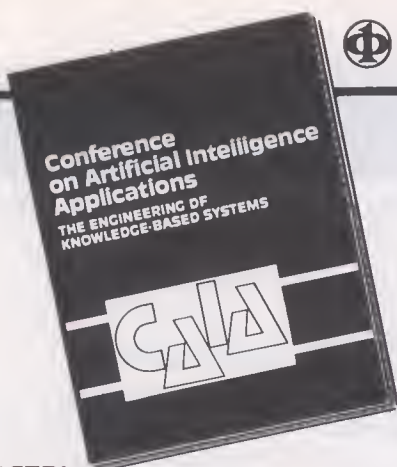
Analog Devices AD684 SHA	A quad sample-and-hold amplifier with max acquisition time per channel of 600 ns ($\pm 0.1\%$) and $1\mu\text{s}$ ($\pm 0.01\%$) for a 10V step. Has independently controlled channels with internal hold capacitors. Comes in a 16-pin skinny DIP in three temperature grades. Cost (100s): starts at \$23.50.	120
Analog Devices ADV476 RAM DAC	A color palette RAM with a triple 6-bit DAC for pixel management and output drive for VGA and IBM PS/2 color graphics. Has a 256×18 color lookup table. Pin- and software-compatible with the Inmos IMSG171. Comes in a 28-pin plastic DIP. Cost (100s): \$16 (35 MHz), \$17 (50 MHz), \$20 (66 MHz).	121
Analog Devices AD1380 ADC	A 16-bit sampling analog-to-digital converter with a max $6\mu\text{s}$ acquisition time and $14\mu\text{s}$ conversion time and a throughput rate of 50 KHz. Comes in a 32-pin triple-width ceramic DIP. Cost (100s): \$126 (JD grade), \$152 (KD grade).	122
Analog Devices AD7846 DAC	A digital-to-analog converter with a segmented ladder architecture for 16 bits of resolution and monotonicity. Includes track-and-hold amplifiers. Comes in a 28-pin plastic or ceramic DIP and 40-terminal LCC. Cost (100s): starts at \$19.	123
Analog Devices AD834 Analog multiplier	A four-quadrant analog multiplier with a 500 MHz min bandwidth and max error of 2%. Available over two temperature ranges: commercial (J grade) and military (S grade). Comes in an 8-pin ceramic DIP or 8-pin small-outline package. Cost (100s): AD834J starts at \$19.20; \$68 for AD834SQ/883B.	124
Chips and Technologies 82C631 EMS mapper chip	A single-chip implementation of the logic required to use the Lotus-Intel-Microsoft Expanded Memory Specification 4.0 in IBM XT, AT, and PS/2 computers. Consists of two register banks of 64 registers. Production scheduled for second quarter. Cost: \$12.50.	125
Fujitsu Microelectronics MB8431, MB8432 SRAMs	Dual-port, 8-bit, slave static RAMs with two independent addressing capabilities, chip selects, write enables, output enables, and I/O ports. Come in 48-pin and 52-pin plastic DIPs and 64-pin plastic flat packs. Cost (1,000s): \$12.	126
Intel 80376 Embedded processor	A 32-bit 80386 microprocessor derivative for embedded applications. Uses a 16-bit external data bus. Delivers 2.5-3.0 MIPS. Now available in production quantities. Comes in 100-pin plastic quad flat pack or 88-lead PGA. Cost (1,000s): \$65.	127
Micro Networks MN5820 Flash ADC	An 8-bit, 20-MHz flash analog-to-digital converter with an output three-state buffer and overrange flag. Contains a band-gap voltage reference, VRT and VRB amps, analog input buffer amp, and input termination resistor. Comes in a 24-pin double-wide DIP. Cost: \$178-\$233.	128
Motorola 88000 Microprocessors	A two-chip family of RISC microprocessors. Available for general sampling, with production scheduled for the second quarter. 88100 CPU integrates integer and floating-point. 88200 CMMU handles cache memory design. Cost (samples): \$494 for 88100, \$619 for 88200.	129
National Semiconductor Military HPC Controllers	Military versions of the 16-bit HPC family. Operate over the -55 to 125°C range. HPC003L17/883 comes in LDCCs and HPC003U17/883 in PGAs. Cost (100s): \$275; \$290 for ROM versions.	130
NCR 86C01 MCA interface	A single-chip Micro Channel bus interface for peripheral chips. Supports 8- or 16-bit data operations, operates in the MCA slave mode, and facilitates DMA data block transfers to and from memory. Cost (1,000s): \$13.95.	131
Texas Instruments TIBPAD18N8-6 PAD	A 6-ns programmable address decoder containing 10 dedicated inputs and 8 product terms, each followed by an inverting buffer leading to an I/O port. Comes in a 20-pin plastic DIP or PLCC. Cost (1,000s): \$9.40.	132
Texas Instruments EP610, EP910 EPLDs	Erasable programmable logic devices, second-source versions of Altera Corp.'s devices. EP610 has 4 dedicated inputs and 16 I/Os. EP910 has 12 dedicated inputs and 24 I/Os. EP610 comes in 24-pin DIPs or 28-pin J-lead LCCs, in 25-, 30-, and 35-ns versions. EP910 comes in 40-pin DIPs or 44-pin J-lead LCC, in 30-, 35-, and 40-ns versions. Cost (1,000s): \$10.78-\$38.08.	133

Microsystem Announcements

Company, Model, Function	Comments	R.S. No.
Force Computers CPU-27 Computer	A 32-bit 68020-based single-board computer that runs at 12.5, 16.7, or 25 MHz. Includes SCSI I/O capability, serial and parallel lines, and optional Ethernet control. Comes with a real-time kernel plus monitor/debugger. Cost: from \$4,490 to \$5,590.	135
Industrial Computer Source B386 CPU Computer	A 32-bit 80386-based CPU card with BIOS stored in ROM, 80387 or Weitek coprocessor support, battery-backed real-time system clock, address-selectable parity-checked system memory with DMA circuitry, 16 programmable interrupts, and 16-bit AT bus interface. Cost: from \$2,445 to \$12,990.	136
Mayze Systems Synchro 1496 Modem	An autodial V32 9.6 Kbps modem with V33, V29, and V22 bis. Transmits at speeds from 1.2 to 14.4 Kbps. Offers 12 preset configurations, stores four user-defined configurations. Monitors the transmission line. Comes as a stand-alone unit or as a card. No price given.	137
Northern Telecom Memorybank 940E Subsystem	A hard-disk/backup tape subsystem for Macintosh networks. Has 940 Mbytes of disk capacity and 2.3 Gbytes of tape backup storage, plus support software for disk management, data backup, and data restoration. Cost: \$17,900.	138
Orchid Technology Privilege 286-20e Motherboard	A motherboard running at 20 MHz using an 80286 from Harris Semiconductor. Employs a passive backplane architecture. Includes up to 8 Mbytes of direct access memory on the backplane. Supports EMS memory, DOS, OS/2, Network, and Unix. Cost (50s): \$463 (0 Kbytes).	139
Paracom MTM-Sun/xp Motherboard	A Transputer-based motherboard, 9U VMEbus compatible, for Sun systems. Features three 96-pin daughterboard connections. Contains four T414 or T800 processors. Can provide up to 10 Transputers. Has a 64 × 64 serial channel Network Configuration Unit on-board. Cost: starts at \$12,895.	140
Paracom BBK-Q Bus interface	A bus bridge for DEC's Q-bus systems and external 32-bit Transputer processors. Adheres to DEC's DR11-W parallel interface standard. Comes with a VAX/VMS V5.0-compatible device driver. Cost: \$2,995.	141
Processor Sciences VGAvision Video board	A video overlay board that adds NTSC video to an IBM PC, XT, AT, or compatible with any VGA board and monitor. Features noninterlaced video output at VGA frequencies. Requires one slot, two I/O ports, and no PC memory space. Uses the ITT Digit 2000 chip set. Cost: \$1,299.	142
Quam Qc 1094 Computer	A single-board computer based on Intel's 80386SX processor. Runs at 16 MHz, accesses up to 8 Mbytes of on-board RAM. Comes with interfaces for floppy and hard disk drives, two serial ports, and one printer port. Cost: \$895 for evaluation card with zero memory.	143
Radstone Technology PIO-1 VME board	A VME board with 32 channels of digital parallel I/O for process control and industrial automation. Operates as an A24-D16 VMEbus slave with interrupt facilities. Uses only the P1 connector. Configurable for input only, output only, or monitored output. Cost: \$890.	144
Rapid Technology ASAP32 Array processor	An array processor board for the IBM PC AT, based on a 32-bit floating-point processor. Includes 2K × 32 bits of ROM and RAM, the ROM with basic math, vector, and I/O functions and the RAM loadable by the host PC with application-specific microcode. Also has a double buffer SRAM, 4K × 32 bits each. Cost: \$1,350.	145
Truevision NuVista HR Color card	A single-slot color display card for the Macintosh II. Provides 1,280 × 960 pixel display using 256 colors from a palette of 16.7 million. Comes in 2- and 4-Mbyte versions. Uses the TI 34010 graphics processor. Compatible with Apple's QuickDraw. Cost: \$3,995 (2 Mbytes) and \$5,995 (4 Mbytes).	146
US Video Recordable VGA Video card	A video card for the IBM PC, XT, AT, or compatible that provides video output for recordability of VGA text and graphics on any VCR. Can also produce an NTSC video signal. Supports 640 × 480 in 256 colors, 800 × 600 in 16 colors, and 1,024 × 768 in 16 colors out of 256,000 displayable. Cost: \$785.	147
Wave Mate Bullet-386SX Motherboard	A motherboard based on the 16-bit, 16-MHz Intel 80386SX CPU. Runs with zero-wait-state RAM. RAM available in 512-, 1,024-, and 4096-Kbyte versions. Features standard AT bus and AT peripherals on the system board. Comes with a RAM utilities package; 80387SX optional. Cost: \$823 (512 Kbytes DRAM).	148



IEEE COMPUTER SOCIETY

THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INCCOMPUTER
SOCIETY
PRESSThe Computer Society Press
presents our newest

ARTIFICIAL INTELLIGENCE PROCEEDINGS

JUST RELEASED!

Proceedings: 5th Conference on Artificial Intelligence Applications

Devoted to the application of artificial intelligence techniques to real-world problems. Focuses on principles which underlie knowledge-based system applications and on case studies of knowledge-based application programs that solve significant problems. Papers focusing on principles of knowledge-based systems describe significant completed research, detailing how relevant AI techniques were applied effectively to important problems, software life cycle concerns, etc. Papers describing case studies describe applications of AI technology, demonstrating the solution of a significant problem, and include analyses of why the implementation techniques selected were appropriate.

Order No. FZ1902

ISBN 0-8186-1902-3; March 1989, 316 pp.

list price \$64.00/member price \$32.00

NEW RELEASES!



NEW!

Proceedings: 1989 Artificial Intelligence Applications in Government Conference

Focuses on the use and development of artificial intelligence systems by government agencies. Topics covered include machine learning, logic-based and common sense reasoning, reasoning under uncertainty, natural language processing, logistics, resource allocation, robotics, autonomous vehicles, advanced architectures, law, planning, and integrating methodologies.

Order No. FZ1934

ISBN 0-8186-1934-1

Mar. 1989, c.400 pp.

list price \$70.00

member price \$35.00

(pre-publication price--subject to change)

NEW EVENT!

Proceedings: Supercomputing '88

Discusses new advances and future directions in the area of supercomputing. Topics include supercomputing execution, development, application, system evaluation, management issues, mass storage, technical aspects of products, and user experience. Geared for all persons connected with supercomputers: manufacturers, suppliers, and users from a wide variety of disciplines.

Order No. FZ882

ISBN 0-8186-0882-X

Nov. 1988, 470 pp.

list price \$80.00

member price \$40.00

NEW EVENT!

Proceedings: Supercomputing '88 -- Science and Applications

Includes papers that were presented at Supercomputing '88 but were not included in the conference proceedings. Edited by Joanne Martin and Stephen F. Lundstrom.

Order No. FZ1923

ISBN 0-8186-9923-X

Mar. 1989, c.270 pp.

list price \$49.95

member price \$34.95

(pre-publication price--subject to change)

ORDER TODAY!

Charge It!

Call toll-free 1-800-CSBOOKS (1-800-272-6627) (in California call 714-821-8380) 8am to 4pm Pacific Time

We accept: VISA • MasterCard • American Express

Please have your IEEE or CS membership number available for member discount.

(Prices outside North America vary. All books subject to availability and prices subject to change.)

Or Write:

U.S. & Canada: Customer Service, IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720 USA

Europe: Customer Service, IEEE Computer Society, 13, Avenue de l'Aquilon, B-1200 Brussels, BELGIUM

Asia & Australia: Customer Service, IEEE Computer Society, Ooshima Building, 2-19-1 Aoyama, Minato-ku, Tokyo 107 JAPAN

Interested in authoring a CS Press book?

Contact: Editor-in-Chief, CS Press, 1730 Massachusetts Avenue, N.W., Washington, DC 20036

CONFERENCES

Department Editor Edmund L. Gallizzi of Eckerd College is serving in a visiting post at CRAI in Italy until July 31, 1989. Contact him at Consorzio per la Ricerca e le Applicazioni di Informatica, 87036 S. Stefano di Rende (CS), Italy; phone 011 (39) 984-833286; electronic mail, gallizzi@crai.uucp

Jacobsen to address IEEE Robotics and Automation Conference

Steve Jacobsen of the University of Utah Center for Engineering Design will deliver a talk entitled "Real Robots Show Just How Bad Reality Is" during the May 17 evening banquet at the 1989 IEEE International Robotics and Automation Conference.

Jacobsen's talk will deal with the next generation of Disney robots and other robot projects featuring multifingered hands and force-reflecting teleoperation arms and hands.

The May 14-19 conference in Scottsdale, Arizona, will feature six days of intensive presentations and discussions of intelligent robot systems. The event is sponsored by the IEEE Robotics and Automation Society.

George A. Bekey of the University of Southern California is the general chair, and John Hollerbach of the Massachusetts

Institute of Technology is the program chair.

The technical paper presentations will cover such topics as architectures, simulators, and systems; artificial intelligence and task planning; control of position and force; dextrous grasping, haptics, and tactile sensing; industrial applications; kinematics and dynamics; locomotion; mechanical design; mobile robots; nonvisual sensing and sensor fusion; path planning and collision avoidance; telerobotics; unstructured environments—space, undersea, nuclear plants; and vision.

Tutorials are slated Sunday, May 14, and the morning of Monday, May 15. The May 14 tutorials are titled "Artificial Intelligence: Techniques and Applications" and "Theory and Application of Redundant Robots," and the May 15

tutorials will cover "Autonomous Mobile Robots" and "Fundamentals of Robot Vision Systems."

In addition, workshops will be devoted to "Task Strategy Generation and Skill Acquisition for Advanced Manipulation" (May 15), "Parallel Algorithms and Architectures in Robotics" (Friday, May 19) and "Integration of AI and Robotic Systems" (also May 19).

Additional information and a copy of the advance announcement can be obtained by telephoning Harry Hayman at (407) 483-3037, contacting Robotics and Automation, PO Box 3216, Silver Spring, MD 20901, or phoning (301) 434-1990 after April 10.

Attendees save \$25 by registering before May 1.

Massively parallel supercomputers hold key to scientific discovery, says Frontiers 88 keynoter

*James R. Fischer, NASA/Goddard Space Flight Center
Mark A. Stevens, ST Systems Corporation*

"The supercomputer is essentially an instrument for scientific discovery," Kenneth G. Wilson, 1982 Nobel Laureate in physics, told his audience when he keynoted the second Symposium on the Frontiers of Massively Parallel Computation.

James R. Fischer of the NASA/Goddard Space Flight Center was general chair of Frontiers 88, with David H. Schaefer of George Mason University serving as program chair. The IEEE Computer Society, the IEEE National Capital Area Council, NASA, and George Mason University sponsored the event, held in Fairfax, Virginia, October 10-12.

Comparing supercomputer development to the evolution of the telescope, Wilson of Ohio State University predicted we will see the supercomputer evolve over the next four centuries just as the microscope and telescope have evolved from the simple optical instruments of Galileo's time four centuries ago to extremely sophisticated systems

such as the Hubble Space Telescope.

In drawing his analogy with the microscope and telescope, Wilson said that data visualization, the eyepiece of the supercomputer, is not yet perfected. He challenged the audience to develop faster, larger supercomputers that can more effectively model and predict complex dynamic systems ranging from Earth science phenomena to the simulation of complex molecules, and to revolutionize the programming environments for these supercomputers in the process.

The symposium sessions responded to Wilson's appeal through 52 talks and 70 poster presentations on computers capable of applying tens of thousands of processing elements to a single job and the data parallel techniques used in their programming.

Echoing the need for larger, more powerful supercomputers, Gary Demos, producer of computer graphics for movies and television, urged the audience to push the state of the art in



Kenneth G. Wilson presented the keynote address at Frontiers 88.

scientific computing.

"Graphics, as a discipline, acts very much like many scientific problems and tends to stress a lot of the general performance capabilities," Demos said. He has many years of experience using supercomputers, including the 65,536-processor Connection Machine-2 to develop high-end graphics algorithms. Demos punctuated his talk with numerous examples of graphics he

developed for movies, such as the design and color techniques used in generating the spaceship scenes for *The Last Starfighter*, which took nine months of solid computation to create.

A panel composed of members of the Frontiers 88 program committee debated the relative futures of single-instruction, multiple-data stream and multiple-instruction, multiple-data stream computers. Most agreed that future massively parallel computers will incorporate features from both architectures to create hybrids that effectively solve a broader range of problems.

The panel agreed that massively parallel computing environments will become comparable to those of vector processors and will not be held back because of programmer resistance to new machines that do not port existing Fortran-based programs.

The panelists represented a cross-section of academia, government, and industry and included Schaefer, Milton Halem (NASA/GSFC), Kai Hwang (University of Southern California), Dennis Parkinson (Active Memory Technology), John Reif (Duke University), Anthony Reeves (Cornell University), Paul Schneck (Supercomputing Research Center), Guy Steele, Jr. (Thinking Machines Corporation), and Leonard Uhr (University of Wisconsin).

During an invited presentation, Azriel Rosenfeld of the University of Maryland discussed the impact massively parallel computers are having on image processing and computer vision. Rosenfeld included examples drawn from his own experience using a CM-2 at the university and expressed his wish for a massively parallel machine with a thousand-dollar price tag.

"It would surprise me if, by the turn of the century, we weren't within shooting distance of that price, if not already there," Rosenfeld said.

Steele, designer of the C* and *Lisp parallel languages implemented on the Connection Machine, discussed common design issues of programming languages during another invited presentation. He examined some of the commonalities of programming languages used on the CM-2, such as C* and Fortran, and stressed the importance of building on existing work done on both serial and parallel computers.

Looking to the future, Steele predicted that languages will have to incorporate increasingly sophisticated functions.

"Themes that will become increasingly common include parallel matrix operations," he said. Steele also men-

tioned sorting and convolution as operations that will play an increasingly important role in future parallel programming languages.

Prior to the technical sessions, a tutorial session examined the architectures, programming languages, and algorithms for SIMD parallel computing systems. Lecturers included Parkinson, who explained the application of massively parallel supercomputers to a wide variety of problems using exam-

ples drawn from his 13 years of experience in programming the Distributed Array Processor.

The symposium proceedings can be obtained by specifying order No. 892 when contacting the Computer Society Press, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-2578, phone (800) CS-BOOKS. In California, dial (714) 821-8380. The IEEE and Computer Society-member price is \$48, and the nonmember price is \$96.

Thompson to deliver keynote at ASPLOS-III

Joel Emer, Digital Equipment

Ken Thompson of AT&T Bell Laboratories, cocreator of the Unix operating system and coauthor of the Belle championship computer chess program, will keynote the third International Conference on Architectural Support for Programming Languages and Operating Systems. The conference will be held in Boston April 3-6.

Joel Emer of DEC is the general chair, and John Hennessy of Stanford University is the program chair. The conference is sponsored by the ACM in cooperation with the IEEE Computer Society's Technical Committees on Microprocessors and Microcomputers, VLSI, and Operating Systems.

In his keynote address, Thompson plans to present his views on the future evolution of computer architecture. A panel session is also planned to discuss the relative merits of contemporary architectural alternatives such as wide-instruction-word machines, superscalar machines, and micropipelined machines.

The conference will begin with two half-day tutorials. Jim Goodman of the University of Wisconsin at Madison will conduct a tutorial on "Cache Consistency and Shared Memory Multiprocessors," and Ron Cytron of IBM's T.J. Watson Research Center will conduct the other on "Program Analysis and Restructuring for Concurrency."

A 2½-day single-track technical program will follow, featuring 27 papers selected from over 100 submittals. Topics to be covered include evaluations of wide-instruction-word architectures, a variety of instruction-level parallelism schemes, cache behavior, architectural support for synchronization, debugging, and specific languages.

An advance program and registration form can be found on pp. 108-109 of the January issue of *Computer*. Further information can be obtained by contacting Toby Bloom at (617) 253-6023.

Moving?

PLEASE NOTIFY
US 4 WEEKS
IN ADVANCE

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

ATTACH
LABEL
HERE

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address below.
- If you have a question about your subscription, place label here and clip this form to your letter.

Name (Please Print)

New Address

City

State/Country

Zip

Bell-Northern's Sakus to keynote Infocom 89

Celia Desmond, Bell Canada

Gedas Sakus, president of Bell-Northern Research, will be the keynote speaker at the eighth annual IEEE Conference on Computer Communications April 23-27. Sakus will speak at 9:30 a.m. Tuesday, April 25.

Slated in Ottawa, Canada, the 1989 conference will be the first in the Infocom series held outside the US. The event is sponsored by the IEEE's Computer and Communications Societies.

Celia Desmond of Bell Canada is serving as the general chair of Infocom 89, with Jon W. Mark of the University of Waterloo the technical chair.

This year's Infocom theme is "Technology: Emerging or Converging?" and the technical program will cover key issues in the computer communications field. Since the first conference in the series in 1982, the number and quality of submitted papers has steadily increased.

The three final conference days will feature some 40 technical sessions, with paper presentations and panel discussions.

The paper sessions will focus on such topics as LANs, MANs, and WANs (local, metropolitan, and wide area networks), gateways/bridges, packet radio networks, packet voice systems, network management design/reliability, security, performance, modelling, interconnection, and architectures.

Panel discussions will focus on "Applications and Services in High-Speed Networks," "ISDN Conformance Testing," "Networks for Intelligent Manufacturing Systems," "Network Security: Present and Future Technology," "User Access to Broadband ISDN: Full ATM or Otherwise?" "Groupware and Telecommunication Management," and "Frame Relaying Service."

To open Infocom 89, well-known researchers will conduct two days of tutorials. Two half-day tutorials will be held Sunday, April 23. One, entitled "SuperNode," will be conducted by John Perry of Bell-Northern Research. The other, titled "ISDN and BISDN Services and Applications," will be conducted by a Bell Communications Research team made up of Stephen B. Weinstein, Joseph E. Rizzo, and Stephen E. Minzer.

Perry's presentation will feature a discussion of digital multiplex systems (DMS). He will address the evolution of a network switching node to an integrated network node.

Weinstein, Rizzo, and Minzer will

discuss the introduction and application of the Integrated Services Digital Network and describe its evolution to broadband ISDN, specifically addressing multimedia, computer-supported applications.

Three full-day tutorials are scheduled Monday, April 24. Ivan P. Kaminow and Nicholas Maxemchuk, two AT&T Bell Laboratories researchers, will discuss optical components and their effects on network topologies and access protocols. Consultant Harold

Podell will focus on network security, emphasizing OSI security architecture and the Trusted Computer System Evaluation Criteria (a US Defense Department standard). Radia Perlman of DEC will address two schemes for interconnecting local area networks.

For an advance program, see pp. 99-102 of the January 1989 issue of *Computer* or contact Celia Desmond, Bell Canada, 438 Bay St., Floor 5, South Tower, Toronto, Ont., Canada M5G 2E1, phone (416) 581-2318.

Innovations for advancement of simulation, simulation applications presented at WSC 88

Carolyn K. Davis, General Dynamics

Robert G. Strader, Stephen F. Austin State University

Keynoter Julian Reitman predicted faster simulation systems and improved user interfaces to assist model development in the not-too-distant future when he spoke at the 1988 Winter Simulation Conference in San Diego.

Peter Haigh of NCR served as WSC 88 conference chair, and John Comfort of Florida International University was the program chair. The IEEE Computer Society and its Technical Committee on Simulation were among seven sponsors of the December 12-14 event.

Reitman, who has a distinguished list of credentials in support of simulation, is affiliated with the University of Connecticut at Stamford and George Mason University. His talk was entitled "A Concise History of the Ups and Down of Simulation."

Explaining that the outlook for simulation is influenced by hardware and software developments, Reitman said that absorbing the rapid hardware changes going on will take time.

But, he said, "When current hardware capabilities are fully used, we will be able to use a network of processors, PC or microprocessor elements, to retain the parallelism of the real world; increase the number of processors to speed up the simulation to meet realistic real-world speed and capacity requirements; include in the processing network some processors using actual real-time data along with those doing the simulation... to allow some elements to be performing discrete events

and others continuous simulations; and dedicate graphical coprocessors to enable the viewer to observe the simulation in the simulation time reference."

He added that "simulation language capabilities will have additional features to reduce the effort to build the model and to allow greater flexibility when running the simulation."

The conference drew over 500 from academia and industry representing a variety of specializations.

WSC's technical program featured tutorials and over 120 paper presentations on methodologies and applications, all designed to provide material for both the novice and the advanced simulationist. The conference provided attendees an opportunity to share new developments and keep updated in many areas of simulation.

The tutorials ranged from general introductory software to specific simulation topics.

The modeling methodology track offered a variety of interests, including knowledge-based simulation and distributed/parallel simulation. The manufacturing and military applications tracks described many application-specific problems the simulationist faces. Some presenters illustrated problems and solution techniques; others demonstrated how new technologies can be applied to improve decision making.

The next conference in the WSC series will be held in Washington, DC, December 4-6.

CALENDAR

March 1989

Ⓜ PCCC 89, Phoenix Conf. on Computers and Communications, Mar. 22-24, Scottsdale, Ariz. Cosponsor: Arizona State Univ. Contact T.D. Regulinski, Reliability Training Inst., PO Box 275, Avondale, AZ 85323, phone (602) 264-2335.

Ⓜ AISIG 89, Artificial Intelligence Systems in Government Conf., Mar. 27-31, Washington, DC. Cosponsors: George Washington Univ., Mitre Corp. Contact Jude Franklin, Government Information Systems, 1500 Planning Research Dr., McLean, VA 22102, phone (703) 556-1990; or IEEE Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013.

SPIE Technical Symp. on Aerospace Sensing, Mar. 27-31, Orlando, Fla. Sponsor: Int'l Society for Optical Engineering. Contact Doris Dowd, SPIE, PO Box 10, Bellingham, WA 98227-0010, phone (206) 676-3290.

First Conf. on Innovative Applications of Artificial Intelligence, Mar. 28-30, Stanford, Calif. Sponsor: American Assoc. for Artificial Intelligence. Contact AAAI, 445 Burgess Dr., Menlo Park, CA 94025, phone (415) 328-3123.

Baltimore 89 Manufacturing Productivity Conf., Mar. 28-30, Baltimore. Sponsor: Society of Manufacturing Engineers. Contact Technical Activities, SME, PO Box 930, Dearborn, MI 48121-0930, phone (313) 271-1080.

Eastern Multiconference, Mar. 28-31, Tampa, Fla. Sponsor: Society for Computer Simulation. Contact SCS, PO Box 17900, San Diego, CA 92117-7900, phone (619) 277-3888; or Allan Rutan, Radar Systems Lab, Raytheon, Wayland, MA 01778, phone (617) 440-5088.

Ⓜ Third Parallel Processing Symp., Mar. 29-31, Fullerton, Calif. Cosponsors: California State Univ., Fullerton et al. Contact Larry Canter, Computer Systems Approach, Inc., 1140 S. Raymond Ave., Suite B, Fullerton, CA 92631, phone (714) 738-3414.

Ⓜ IEEE Built-In Self-Test Workshop, Mar. 29-31, Charleston, S.C. Contact Richard Sedmak, Self-Test Services, 6 Lindenwold Terr., Ambler, PA 19002, phone (215) 628-9700.

Eighth Symp. on Principles of Database Systems, Mar. 29-31, Philadelphia. Sponsor: ACM. Contact Avi Silberschatz, Computer Science Dept., Univ. of Texas at Austin, Austin, TX 78712.

Ⓜ Conferences that the IEEE Computer Society sponsors or participates in are indicated by the Computer Society logo; additional conference sponsors are also listed. Other conferences of interest to our readers are included, as well.

For inclusion in Call for Papers or Calendar, submit information six weeks before the month of publication (i.e., for the May 1989 issue, send information for receipt by March 15, 1988) to Chuck Governale, Calendar Dept., *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

Ⓜ Workshop on Applied Computing 89, Mar. 30-31, Stillwater, Okla. Cosponsors: National Science Foundation et al. Contact Donald D. Fisher, Oklahoma State Univ., CIS, Stillwater, OK 74078, phone (405) 624-5668.

CMC 89, Colorado Microelectronics Conf., Mar. 30-31, Colorado Springs, Colo. Contact Conf. Secretary, CMC 89, Microelectronics Research Labs, Univ. of Colorado, Box 7150, Colorado Springs, CO 80933-7150, phone (719) 593-3488.

Western Educational Computing Conf., Mar. 30-31, Santa Cruz, Calif. Sponsor: California Educational Computing Consortium. Contact Judah Rosenwald, Extended Education, NAD 153, San Francisco State Univ., 1600 Holloway, San Francisco, CA 94132, phone (415) 338-1212.

April 1989

Second National Conf. on Telecommunications, Apr. 2-5, York, England. Sponsor: Institution of Electrical Engineers. Contact IEE Conf. Services, Savoy Pl., London WC2R 0BL, UK, phone 44 (1) 240-1871, ext. 222.

Ⓜ ASPLOS III, Third Int'l Conf. on Architectural Support for Programming Languages and Operating Systems, Apr. 3-6, Boston. Cosponsor: ACM. Contact Joel Emer, DEC/MIT, 545 Technology Square (NE43-503), Cambridge, MA 02139, phone (617) 253-7341.

Working Conf. on Visual Database Systems, Apr. 3-7, Tokyo. Cosponsors: IFIP, IPSJ. Contact IFIP TC-2 Working Conf., Tosiya L. Kunii, Information Science Dept., Faculty of Science, Univ. of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan, phone 81 (3) 812-2111, ext. 4116.

Third Eurographics Workshop on Intelligent CAD Systems, Apr. 3-7, Texel, The Netherlands. Sponsor: Center for Mathematics and Computer Science. Contact Marja Hegt, CMCS, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands, phone 31 (20) 592-4058.

Second Florida Artificial Intelligence Research Symp., Apr. 3-7, Orlando, Fla. Contact Avelino J. Gonzalez, Computer Engineering Dept., Univ. of Central Florida, Orlando, FL 32816, phone (407) 281-5027.

Ⓜ Int'l Symp. on Database Systems for Advanced Applications, Apr. 10-12, Seoul, Korea. Cosponsors: IPSJ, Korean Information Science Society. Contact Sukho Lee, Computer Engineering Dept., Seoul National Univ., Sinlim-Dong, Gwanak-ku, Seoul 151, Korea, phone 82 (2) 886-0101.

MIV 89, Int'l Workshop on Industrial Applications of Machine Intelligence and Vision, Apr. 10-12, Tokyo. Cosponsors: IEEE et al. Contact Mitsuru Ishizuka, Inst. of Industrial Science, Univ. of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo, 106, Japan, phone 81 (03) 470-5389.

First Hong Kong Int'l Computer Conf., Apr. 10-14, Hong Kong. Sponsor: American Federation of Information Processing Societies et al. Contact George R. Eggert, AFIPS, 2200E Devon Ave., Suite 268, Des Plaines, IL 60018, phone (312) 299-4227; or Alex Tzang, Suite 705 East Town Bldg., 41 Lockhart Rd., Hong Kong, phone 852 (5) 286-136.

Ⓜ 1989 IEEE VLSI Test Workshop, Apr. 11-13, Atlantic City, NJ. Sponsors: IEEE Computer Society Test Technology Committee, IEEE Philadelphia Section. Contact Wesley E. Radcliffe, IBM-CTD, B/321-5E1, D/277, East Fishkill Facility, Hopewell Junction, NY 12533, phone (914) 894-4346.

Electro 89, Apr. 11-13, New York City. Sponsors: IEEE, ERA. Contact Electro 89, 8110 Airport Blvd., Los Angeles, CA 90045, phone (213) 772-2965.

Ⓜ ETC 89, First European Test Conf., Apr. 12-14, Paris. Cosponsor: Societe des Electriciens et des Electroniciens. Contact Roger Cogonen, 36 Ave. Jean Janurs, 95230 Soisy Sous, Montmorency, France, phone 33 (1) 39-89-03-46; or Colin Maunder, British Telecom Research Labs, Martlesham Heath, Ipswich, Suffolk IP5 7RE, phone 44 (473) 642-706.

NCGA 89, Apr. 17-20, Philadelphia. Sponsor: National Computer Graphics Assoc. Contact NCGA, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031, phone (800) 225-6242.

 **First Symp. on Parallel and Distributed Processing, Apr. 20-21**, Dallas. Contact Behrooz Shirazi, Computer Science Dept., Southern Methodist Univ., Dallas, TX 75275, phone (214) 692-2874.

Multimedia 89, Second Comsoc Int'l Multimedia Communications Workshop, Apr. 20-23, Ottawa, Canada. Cosponsors: IEEE et al. Contact Ottawa Carleton Research Inst., 300 March Rd., Suite 204, Kanata, Ont., Canada K2K 2E2, phone (613) 592-8160.

 **Infocom 89, Conf. on Computer Communications, Apr. 23-27**, Ottawa, Canada. Contact Celia Desmond, Telecom Canada, 438 Bay St., FL5S (C5), South Tower, Toronto, Ont., Canada, M5G 2E1, phone (416) 581-2318.

Vision 89, Apr. 25-27, Chicago. Sponsor: Society of Manufacturing Engineers. Contact Maria Nowakowski, SME, 1 SME Dr., PO Box 930, Dearborn, MI 48121, phone (313) 271-1500, ext. 376.

Third Workshop on Empirical Studies of Programmers, Apr. 29-30, Austin, Tex. Sponsor: Foundation for the Empirical Studies of Programmers. Contact Gary Olson, Cognitive Science and Machine Intelligence Lab, Univ. of Michigan, Ann Arbor, MI 48109-1234, phone (313) 747-4948; or Elliot Soloway, Univ. of Michigan, EECS Dept., 1101 Beal Ave., Ann Arbor, MI 48109, phone (313) 936-1562.

39th IEEE Vehicular Technology Conf., Apr. 29-May 2, San Francisco. Contact Frank Thatcher, 564 Market St., Suite 612, San Francisco, CA 94104, phone (415) 956-6118.

Physical Design Workshop, Apr. 30-May 3, Long Beach, Calif. Sponsors: IEEE, ACM. Contact Jamie Kirkendall, ACM, 11 W. 42nd St., New York, NY 10036, phone (212) 869-7440; or Bryan Preas, Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304, phone (415) 494-4845.

 **CHI 89, Conf. on Human Factors in Computing Systems, Apr. 30-May 4**, Austin, Tex. Cosponsors: ACM, Human Factors Society. Contact Claudia Raun or Bill Curtis, MCC, PO Box 200195, Austin, TX 78720, phone (512) 338-3798.

34th Int'l Instrumentation Symp., Apr. 30-May 4, Orlando, Fla. Sponsor: Instrument Society of America. Contact Frederick A. Kern, PO Box 65, Seaford, VA 23696, phone (804) 865-3269.

May 1989

 **1989 IEEE Symp. on Research in Security and Privacy, May 1-3**, Oakland, Calif. Contact Terry V. Benzel, Trusted Information Systems, 11340 Olympic Blvd., Suite 265, Los Angeles, CA 90064, phone (213) 477-5828.

1989 SME Int'l Conf., May 1-4, Detroit. Sponsor: Society of Manufacturing Engineers. Contact SME, PO Box 930, Dearborn, MI 48121, phone (313) 271-1500.

Third AI Research in Environmental Science Workshop, May 2-4, Washington, DC. Contact William R. Moninger, Environmental Research Labs, NOAA, R/E2, 325 Broadway, Boulder, CO 80803, phone (303) 497-6435.

Sixth Canadian Symp. on Instructional Technology, May 3-5, Halifax, N.S., Canada. Sponsor: National Research Council Canada. Contact Laurier Forget, CSIT, Conf. Services, NRCC, Ottawa, Ont. K1A 0R6, Canada, phone (613) 993-9009.

20th Pittsburgh Conf. on Modeling and Simulation, May 4-5, Pittsburgh, Pa. Sponsor: Univ. of Pittsburgh. Contact William G. Vogt or Marlin H. Mickle, 348 Benedum Engineering Hall, Univ. of Pittsburgh, Pittsburgh, PA 15261.

Robots 13, May 7-11, Gaithersburg, Md. Sponsor: Society of Manufacturing Engineers. Contact Rebecca Alsop, SME, 1 SME Dr., Dearborn, MI 48121, phone (313) 271-1500, ext. 358.

STA 5, Fifth Structured Techniques Assoc. Conf., May 8-11, Chicago. Sponsors: STA, ACM. Contact STA, c/o Robert Binder Systems Consulting, Inc., 3 First National Plaza, Suite 1400, Chicago, IL 60602.

34th Int'l SAMPE Symp., May 8-11, Reno, Nev. Sponsor: Society for the Advancement of Material and Process Engineering. Contact Marge Smith, SAMPE, PO Box 2459, 843 W. Glentana, Covina, CA 91722, phone (818) 331-0616.

 **CompEuro 89, Int'l Conf. on VLSI and Computer Peripherals, May 8-12**, Hamburg. Cosponsors: Gesellschaft für Informatik et al. Contact Walter E. Proebster, IBM Lab, PO Box 1380, D-7030 Boebblingen, Schonaicher Str. 220, Federal Republic of Germany, phone 49 (70) 3116-3929.

ICCAL 89, Second Int'l Conf. on Computer-Assisted Learning, May 9-11, Dallas. Sponsor: Computer Learning Research Center, Univ. of Texas at Dallas. Contact Janet Harris, Center for Continuing Education, Univ. of Texas at Dallas, PO Box 830688, MS CN 1.1, Richardson, TX 75083-0688.

CCC 89, Second Hungarian Custom Circuit Conf., May 10-12, Szeged, Hungary.

Cosponsors: Scientific Society of Measurement and Automation (MATE). Contact MATE Secretariat, 1055 Budapest, Kossuth L. tér 6-8, Hungary, phone (1) 531-406.

 **Sixth IEEE Workshop on Real-Time Operating Systems and Software, May 11-12**, Pittsburgh. Cosponsor: Carnegie Mellon Univ. Contact Andre van Tilborg, Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217-5000, phone (202) 696-4302.

AAMSI Congress 89, May 11-13, San Francisco. Sponsor: American Assoc. for Medical Systems and Informatics. Contact AAMSI, Suite 700, 1101 Connecticut Ave. NW, Washington, DC 20036, phone (202) 857-1189.

First Int'l Workshop on Human and Machine Cognition, May 11-13, Santa Rosa Island, Fla. Contact Ken Ford, Computer Science Div., Univ. of West Florida, Pensacola, FL 32514, phone (904) 474-2551.

Int'l Conf. on Robotics and Automation, May 14-19, Scottsdale, Ariz. Sponsor: IEEE. Contact Harry Hayman, PO Box 3216, Silver Spring, MD 20901, phone (301) 434-1990 or (407) 483-3037.

 **11th Int'l Conf. on Software Engineering, May 15-18**, Pittsburgh. Cosponsor: ACM. Contact Larry Druffel, Software Engineering Inst., Carnegie Mellon Univ., Pittsburgh, PA 15233, phone (412) 268-7740; or IEEE Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-0101.

Sixth Conf. on Real-Time Applications in Nuclear, Particle, and Plasma Physics, May 15-18, Williamsburg, Va. Sponsors: IEEE et al. Contact Roy Whitney, 12000 Jefferson Ave., Newport News, VA 23606, phone (804) 249-7536.

18th Mumps Users Group Annual Meeting, May 15-19, Seattle. Contact Mumps Users Group, 4321 Hartwick Rd., Suite 510, College Park, MD 20740, phone (301) 779-6555.

SID 89, Society for Information Display Int'l Symp., Seminar, and Exhibition, May 15-19, Baltimore. Contact Society for Information Display, c/o Palisades Inst. for Research Services, 201 Varick St., Rm. 1140, New York, NY 10014, phone (212) 620-3375.

 **Int'l Symp. on VLSI Technology, Systems, and Applications, May 17-19**, Taipei, Taiwan. Cosponsors: Republic of China National Science Council, Industrial Technology Research Inst. Contact Alice M. Chiang, MIT, Lincoln Lab, Lexington, MA 02173-0073, phone (617) 981-4629.

Chapel Hill Workshop on Volume Visualization, May 18-19, Chapel Hill, N.C. Sponsor: Univ. of North Carolina, Molecular Graphics Society. Contact Frederick P. Brooks, Jr., Computer Science Dept., Univ. of North Carolina, Chapel Hill, NC 27599-3175.

 **Fifth Int'l Workshop on Software Specification and Design, May 19-20,** Pittsburgh. Cosponsor: ACM. Contact Sol J. Greenspan, GTE Labs, 40 Sylvan Rd., Waltham, MA 02254, phone (617) 466-2962; or Colin Potts, MCC, 9390 Research Blvd., Kaleido II Bldg., Austin, TX 78759, phone (512) 338-3629.

 **First IEEE Symp. on Parallel and Distributed Processing, May 22-23,** Dallas. Sponsor: Dallas Section of the IEEE Computer Society. Contact Mark Shadown, Information Technologies Lab, Texas Instruments, PO Box 655474, MS 238, Dallas, TX 75265, or call Behrooz Shirazi at (214) 692-2874.

AMAST, Int'l Conf. on Algebraic Methodology and Software Technology, May 22-24, Iowa City, Iowa. Contact Eugene Madison or Teodor Rus, Computer Science and Mathematics Dept., Univ. of Iowa, Iowa City, IA 52242, phone (319) 335-0742 or 0694.

Sixth Int'l Conf. on Testing Computer Software, May 22-25, Washington, DC. Sponsors: DPMA, ACM. Contact Genevieve Houston-Ludlam, Frontier Technologies, 2444 Solomons Island Rd., Suite 205, Annapolis, MD 21401, phone (301) 266-8244.

 **10th Tunisian French Seminar of Computer Science: The Role of Languages in Programming, May 23-25,** Tunis, Tunisia. Cosponsor: Tunisian Information Processing Society. Contact Abdelfettah Belghith, Dept. d'Informatique, Faculte des Sciences de Tunis, 1002 Belvedere Tunisia; or Ali Mili, Faculty of Sciences, Univ. of Tunis II, Campus Universitaire, 1002 Belvedere, Tunisia.

SIGMetrics 89 and Performance 89, May 23-26, Berkeley, Calif. Sponsors: ACM, IFIP. Contact Luis-Felipe Cabrera, IBM Almaden Research Center, Mail Code K52/803, San Jose, CA 95120-6099, phone (408) 927-1838.

ICCI 89, Int'l Conf. on Computing and Information, May 23-27, Toronto. Contact Waldemar W. Koczkodaj, Laurentian Univ., CoSc, Sudbury, Ont., Canada P3E 2C6, phone (705) 675-1151.

Workshop on New Directions in Computer Chess, May 28-June 1, Edmonton, Alta., Canada. Sponsors: Int'l Computer Chess Assoc., Canadian Information Processing Society. Contact Tony Marsland, Computing Science Dept., Univ. of Alberta, Edmonton, Alta., Canada T6G 2H1.

 **16th Int'l Symp. on Computer Architecture, May 28-June 1,** Jerusalem, Israel. Cosponsor: ACM. Contact M. Yoeli, Computer Science Dept., Technion City, Haifa 32000, Israel, phone 972 (4) 294-314.

 **19th Int'l Symp. on Multiple-Valued Logic, May 29-31,** Guangzhou, China. Cosponsors: Chongqing Univ. et al. Contact David M. Miller, Computer Science Dept.,

WHERE SOFTWARE IS A SCIENCE

TASC is a highly-regarded and rapidly-growing 1500-person applied research organization, providing government and commercial clients with a broad spectrum of computer-based analytic services and software. Here, computer science professionals play key roles in a variety of important technical activities within such fields as imaging, artificial intelligence and advanced computer graphics. We currently have several opportunities for successful computer scientists who can meet TASC's standards, widely-acknowledged to be challenging, demanding and rewarding. If you are interested in learning more about our unique and successful environment, where software is indeed a science, please review the following openings:

IMAGE SYSTEMS ENGINEER

Apply image processing, computer vision and AI techniques within such fields as remote sensing, target detection, data fusion and decision support. Requires an MS in EE, CS or a related field and 5 years' professional and/or academic background in appropriate activities. Publications and/or management experience are desirable.

GRAPHICS/IMAGE SCIENTIST

Take a senior technical role in all phases of graphics systems development and analysis, utilizing expertise in synthetic scene generation, computer graphics and sensor simulation. Requires MS in a technical field, and experience in graphics software development for 3-D modeling and animation. Familiarity with SUN, PIXAR and Silicon Graphics hardware desirable, as is experience with parallel/supercomputing technologies.

GRAPHICS PROGRAMMER

Develop concepts, algorithms, systems and graphical techniques for visual simulation, graphics interfaces and scientific visualization. Requires BS in a technical field; and experience in developing graphics software for 3-D modeling and animation with SUN, PIXAR and Silicon Graphics hardware a plus.

TASC provides a unique compensation plan, including a salary philosophy sensitive to achievement, a supportive program of benefits, and immediate participation in a well-established profit-sharing plan.

Please write to Coley I. Rybicki.

**An Equal Opportunity
Employer, M/F.
U.S. Citizenship
Required.**

TASC
THE ANALYTIC SCIENCES CORPORATION

55 Walkers Brook Drive
Reading, MA 01867

Univ. of Victoria, B.C., Canada, V8W 2Y2, phone (604) 721-7220.

CIPS National Congress 89, May 29-June 2, Edmonton, Alta., Canada. Sponsor: CIPS. Contact Congress 89, PO Box 1277, Main Post Office, Edmonton, Alta., Canada T5J 2M8, phone (403) 488-1879

 **Int'l Conf. on Systolic Arrays, May 31-June 2,** Killarney, Ireland. Cosponsor: Queen's Univ. of Belfast. Contact Earl Swartzlander, TRW, 1 Space Park, Redondo Beach, CA 90278, phone (213) 535-4321.

June 1989

IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing, June 1-2, Victoria, B.C., Canada. Sponsor: IEEE Victoria Section, Univ. of Victoria. Contact Warren D. Little, Dept. of Electrical and Computer Engineering, Univ. of Victoria, PO Box 1700, Victoria, B.C., Canada V8W 2Y2, phone (604) 721-8686.

ICGA 89, Third Int'l Conf. on Genetic Algorithms, June 4-7, Washington, DC. Contact J. David Schaffer, Philips Labs, 345 Scarborough Rd., Briarcliff Manor, NY.

 **CVPR 89, Conf. on Computer Vision and Pattern Recognition, June 4-8,** San Diego, Calif. Contact Rama Chellappa, PHE324, Electrical Engineering-Systems Dept., Univ. of Southern California, University Park, MC-0272, CA 90089, phone (213) 743-8559.

 **Fourth Israel Conf. on Computer Systems and Software Engineering, June 5-6,** Tel Aviv. Cosponsors: Israel Chapter of IEEE Computer Society, Information Processing Assoc. of Israel. Contact S. Koenig, Ortra Ltd., PO Box 50432, Tel Aviv 61500, Israel, phone 972 (3) 664-825.

 **Fourth Symp. on Logic in Computer Science, June 5-8,** Pacific Grove, Calif. Cosponsor: ACM. Contact Albert Meyer, Computer Science Lab, MIT, 545 Technology Square, Cambridge, MA 02139.

 **Ninth Int'l Conf. on Distributed Computing Systems, June 5-9,** Newport Beach, Calif. Contact Kane Kim, Computer Engineering Program, Electrical Engineering Dept., Univ. of California, Irvine, CA 92717, phone (714) 856-5542.

 **Working Conf. on Computer-Aided Design Systems Using Artificial Intelligence Techniques, June 6-7,** Tokyo. Cosponsor: IFIP, IPSJ. Contact Akihiko Yamada, NEC, 4-14-22 Shibaura, Minato-ku, Tokyo 108, Japan; or Kozo Kinoshita, Faculty of Integrated Arts and Sciences, Hiroshima Univ., 1-1-89 Sendamachi, Naka-ku, Hiroshima 730, Japan, phone 81 (82) 249-9150.

Third Int'l Workshop on Wafer Scale Integration, June 6-8, Como, Italy. Sponsor: IFIP. Contact Mariagiovanna Sami, Dip. di Elettronica, Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133 Milan, Italy, phone 39 (2) 23-99-35-16.

 **Second Int'l Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, June 6-9,** Tullahoma, Tenn. Cosponsors: ACM, Univ. of Tennessee Space Inst. Contact Moonis Ali, Univ. of Tennessee Space Inst., Tullahoma, TN 37388, phone (615) 455-0631.

Ninth Int'l Symp. on Protocol Specification, Testing, and Verification, June 6-9, Enschede, The Netherlands. Sponsor: IFIP. Contact ICSC, Univ. of Twente, PO Box 217, 7500 AE Enschede, The Netherlands.

Eighth Biennial University/Government/Industry Microelectronics Symp., June 12-14, Westborough, Mass. Sponsors: Int'l Society for Hybrid Microelectronics, IEEE. Contact Richard B. Gold, Massachusetts Microelectronics Center, 75 North Dr., Westborough, MA 01581.

Workshop on Automatic Verification Methods for Finite State Systems, June 12-14, Grenoble, France. Sponsor: C-cube, the French National Project on Parallelism. Contact Edmund M. Clarke, Jr., Carnegie Mellon Univ., Computer Science Dept., Pittsburgh, PA 15213-3890; A. Pnueli, Weizmann Inst., Rehovot, Israel; or J. Sifakis, LGI-IMAG, BP 53X, 38041 Grenoble Cedex, France.

ICAIL 89, Second Int'l Conf. on Artificial Intelligence and Law, June 13-16, Vancouver, Canada. Contact Robert T. Franson or Joseph C. Smith, Faculty of Law, Univ. of British Columbia, Vancouver, B.C., Canada, phone (604) 228-2323.

Int'l Conf. on Software Engineering and Knowledge Engineering, June 15-17, Chicago. Contact Shi-Kuo Chang, Computer Science Dept., Univ. of Pittsburgh, 322 Alumni Hall, Pittsburgh, PA 15260, phone (412) 624-8490.

First Symp. on Parallel Algorithms and Architectures, June 18-21, Santa Fe, N.M. Sponsor: ACM. Contact Tom Leighton, Math Dept. and Computer Science Lab, MIT, Cambridge, MA 02139.

 **Int'l Workshop on Hardware Fault Tolerance in Multiprocessors, June 19-20,** Urbana, Ill. Contact Prith Banerjee, Coordinated Science Lab, Univ. of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 333-6564.

 **CHDL 89, Ninth Int'l Symp. on Computer Hardware Description Languages and Applications, June 19-21,** Washington, DC. Cosponsors: IFIP, ACM. Contact John A. Darringer, IBM T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598, phone (914) 945-1018.

Sixth Int'l Workshop on Database Machines, June 19-21, Deauville, France. Sponsors: INRIA, AFCET. Contact Haran Boral, MCC, 3500 W. Balcones Center Dr., Austin, TX 78759 (in the US); or Pascal Faudemay, Masi Labo., Univ. of Paris 6, 4 Place Jussieu, 75252 Paris, Cedex 05, France (outside the US).

ICNN 89, Int'l Conf. on Neural Networks, June 19-22, Washington, DC. Sponsor: IEEE. Contact Nomi Feldman, 3770 Tansy St., San Diego, CA 92121, phone (619) 453-6222.

 **Fourth Structure in Complexity Theory Conf., June 19-22,** Eugene, Ore. Cosponsors: Univ. of Oregon, ACM. Contact Stephen R. Mahaney, Rm. 2C-454, AT&T Bell Labs, 600 Mountain Ave., Murray Hill, NJ 07974.

Graphics Interface 89, June 19-23, London, Ont., Canada. Sponsor: Canadian Man-Computer Communications Society. Contact Irene Gargantin, Computer Science Dept., Univ. of Western Ontario, London, Ont., Canada N6A 5B7, phone (519) 661-3563.

Vision Interface 89, June 19-23, London, Ont., Canada. Sponsor: Canadian Image Processing and Pattern Recognition Society. Contact Irene Gargantin, Computer Science Dept., Univ. of Western Ontario, London, Ont., Canada N6A 5B7, phone (519) 661-3563.

 **NECC 89, 10th National Educational Computing Conf., June 20-22,** Boston. Cosponsor: Int'l Council for Computers in Education. Contact Michael C. Mulder, Applied Research, Univ. of Portland, 5000 N. Willamette Blvd., Portland, OR 97203, phone (503) 283-7433; or NECC 89, ICCE, Univ. of Oregon, 1787 Agate St., Eugene, OR 97403-9905.

3rd Electronics Materials and Process Conf., June 20-22, Los Angeles. Sponsor: Society for the Advancement of Material and Process Engineering. Contact Marge Smith, SAMPE, PO Box 2459, 843 W. Glentana, Covina, CA 91722, phone (818) 331-0616.

 **FTCS 19, Int'l Fault-Tolerant Computing Symp., June 21-23,** Chicago. Contact S.M. Reddy, FTCS 19, Electrical and Computer Engineering Dept., Univ. of Iowa, Iowa City, IA 52242, phone (319) 335-5196; or Ravi K. Iyer, Coordinated Science Lab, Univ. of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 333-9732.

SIGPlan 89, Conf. on Programming Language Design and Implementation, June 21-23, Portland, Ore. Sponsor: ACM. Contact Bruce Knobe, Prime Computer, Inc., 500 Old Connecticut Path, Framingham, MA 01701, phone (508) 879-2960.

 **CAR 89, Computer-Assisted Radiology Conf., June 25-28,** Berlin. Cosponsor: AMK Berlin. Contact Michael L. Rhodes, MPD1, 2730 Pacific Coast Hwy., Torrance, CA 90505, phone (213) 539-5944.

CALL FOR PAPERS

Hot Chips Symp.: June 26-27, 1989, Palo Alto, Calif. Sponsor: IEEE Computer Society Technical Committee on Microprocessors and Microcomputers. Submit presentation proposal to Jack Grimes, Intel, SC4-40, 3065 Bowers Ave., Santa Clara, CA 95051; or Dave Ditzel, Sun Microsystems, A7-41, 2550 Garcia Ave., Mountain View, CA 94043.

First Hi-Tech Service and Maintenance Conf.: Oct. 31-Nov. 2, 1989, Chicago. Submit abstract to Donald F. Blumberg, 1260 Virginia Dr., Suite 200, Fort Washington, PA 19034, phone (215) 643-9060.

Int'l J. Computer Aided VLSI Design plans a special issue of neural networks. Publisher: Ablex Publishing. Submit paper by **Mar. 28, 1989**, to Omid Omidvar, Computer Science Dept., Univ. of the District of Columbia, 4200 Connecticut Ave. NW, Washington, DC 20008, phone (202) 282-7345.

First Int'l Workshop on Human and Machine Cognition: May 11-13, 1989, Santa Rosa Island, Fla. Submit paper by **Mar. 30, 1989**, to Ken Ford, Computer Science Div., Univ. of West Florida, Pensacola, FL 32514, phone (904) 474-2551.

Int'l Assoc. of Knowledge Engineers Conf.: June 26-28, 1989, College Park, Md. Cosponsors: IAKE, Univ. of Maryland. Submit paper by **Mar. 31, 1989**, to Michael Teague, IAKE, Georgetown, PO Box 25461, Washington, DC 20007.

Eighth Symp. on Reliable Distributed Systems: Oct. 10-12, 1989, Seattle. Submit paper by **Apr. 1, 1989**, to Raif Yanney, TRW, 1 Space Park, DH2-2328, Redondo Beach, CA 90278, phone (213) 812-6033.

WSC 89, Winter Simulation Conf.: Dec. 4-6, 1989, Washington, DC. Cosponsors: Society for Computer Simulation et al. Submit proposal for session, panel discussion, and tutorial by **Apr. 1, 1989**, and paper by **May 1, 1989**, to Philip Heidelberger, IBM Research Div., T.J. Watson Research Center, Hawthorne, PO Box 704, Yorktown Heights, NY 10598, phone (914) 789-7156.

I-DEAS/CAEDS Users Conf.: Oct. 3-5, 1989, Cincinnati, Ohio. Sponsor: Structural Dynamics Research Corp. Submit abstract by **Apr. 1, 1989**, to Mary Ward, SDRC, 2000 Eastman Dr., Milford, OH 45150, phone (513) 576-2400.

Symp. on Design and Implementation of Large Spatial Databases: July 17-18, 1989, Santa Barbara, Calif. Submit paper by **Apr.**

1, 1989, to Oliver Gunther, Computer Science Dept., University of California at Santa Barbara, Santa Barbara, CA 93106, phone (805) 961-3236.

Seventh Pacific Northwest Software Quality Conf.: Sept. 11-12, 1989, Portland, Ore. Submit abstract by **Apr. 3, 1989**, to Dick Hamlet, Computer Science Dept., Portland State Univ., PO Box 751, Portland, OR 97207-0751.

CASE 89, Third Int'l Workshop on Computer-Aided Software Engineering: July 17-21, 1989, London. Cosponsors: Imperial College of London et al. Submit position paper by **Apr. 7, 1989**, to Elliot Chikofsky, Index Technology Corp., 1 Main St., Cambridge, MA 02142 (for North America); or John Jenkins, School of Management, Imperial College, London SW7 2PG, UK (for other regions).

Int'l Symp. on Computer Architecture and Digital Signal Processing: Oct. 11-14, 1989, Hong Kong. Sponsor: IEEE. Submit paper by **Apr. 7, 1989**, to W.C. Siu, Electronic Engineering Dept., Hong Kong Polytechnic, Hong Kong, phone (852) 3-638344, ext. 496.

Second IEEE Workshop on Workstation Operating Systems: Sept. 27-29, 1989, Pacific Grove, Calif. Submit position statement by **Apr. 15, 1989**, to Luis-Felipe Cabrera, Mail Code K52/803, IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120-6099.

Eighth Int'l Conf. on Entity-Relationship Approach: Oct. 18-20, 1989, Toronto. Sponsor: ER Inst. Submit paper by **Apr. 15, 1989**, to Frederick H. Lochovsky, Computer Science Dept., Univ. of Toronto, Stanford Fleming Bldg., 10 King's College Circle, Toronto, Ont. M5S 1A4, Canada, phone (416) 978-7441.

Micro 22, 22nd Workshop on Microprogramming and Microarchitecture: Aug. 14-16, 1989, Dublin, Ireland. Submit paper by **Apr. 15, 1989**, to Gearold R. Johnson, Center for Computer Assisted Engineering, Colorado State Univ., Fort Collins, CO 80523, phone (303) 491-5543.

Tencon 89, IEEE Region 10 Conf.: Nov. 22-24, 1989, Bombay, India. Sponsor: IEEE Bombay Section. Submit extended abstract by **Apr. 15, 1989**, and paper by **July 15, 1989**, to A.L. Lakshminarasimhan, EECS Dept., Stevens Inst. of Technology, Hoboken, NJ 07030, phone (210) 420-5611.

13th Western Educational Computing Conf.: Nov. 16-17, 1989, Burlingame, Calif. Spon-

sor: California Educational Computing Consortium. Submit abstract by **Apr. 21, 1989**, to Oliver Seely, California State Univ. at Dominguez Hills, 1000 E. Victoria St., Carson, CA 90747, phone (213) 516-3383.

Third Int'l Workshop on Distributed Algorithms: Sept. 26-28, 1989, Nice, France. Submit paper by **Apr. 25, 1989**, to Jean-Claude Bermond, LRI, Bat 490, Universite Paris-Sud, 91405 Orsay, France; or Michel Raynal, IRISA, Campus de Beaulieu, 35042 Rennes, France.

ISCIS 4, Fourth Int'l Symp. on Computer and Information Sciences: Oct. 30-Nov. 1, 1989, Cesme, Turkey. Submit paper by **Apr. 25, 1989**, to Asuman Dogac, Computer Engineering Dept., Middle East Technical Univ., 06531 Ankara, Turkey.

ICCAD 89, IEEE Int'l Conf. on Computer-Aided Design: Nov. 6-9, 1989, Santa Clara, Calif. Cosponsor: IEEE Circuits and Systems Society. Submit paper by **Apr. 28, 1989**, to ICCAD 89 Secretary, IEEE Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013.

Int'l J. Computer Aided VLSI Design plans a special issue on design simulation and implementation. Publisher: Ablex Publishing Corp. Submit paper by **Apr. 30, 1989**, to Harry W. Tyrer, Electrical and Computer Engineering Dept., University of Missouri at Columbia, Columbia, MO 65211, phone (314) 882-6489.

Second Int'l Workshop on Software Configuration Management: Oct. 24-27, 1989, Princeton, NJ. Cosponsors: ACM, Gesellschaft für Informatik. Submit paper by **Apr. 30, 1989**, to Walter Tichy, Univ. of Karlsruhe, PO Box 6980, D-7500 Karlsruhe 1, FRG, phone 49 (721) 608-3934 (in Africa, Asia, and Australia); or Peter Feiler, Software Engineering Inst., Carnegie Mellon Univ., Pittsburgh, PA 15213 (in North and South America), phone (412) 268-7790.

Supercomputing 89: Nov. 13-17, 1989, Reno, Nev. Cosponsor: ACM. Submit paper by **May 1, 1989**, to Gary Johnson, San Diego Supercomputer Center, PO Box 85608, San Diego, CA 92138, phone (619) 534-5181.

10th Real-Time Systems Symp.: Dec. 5-7, 1989, Los Angeles. Submit paper by **May 1, 1989**, to Douglass Locke, IBM Systems Integration Div., MD 0212, Bodle Hill Rd., Oswego, NY, 13827, phone (607) 751-4291.

Third Int'l Workshop on Petri Nets and Performance Models: Dec. 11-13, 1989, Kyoto, Japan. Cosponsors: Society of Instrument and Control Engineers et al. Submit paper by **May 1, 1989**, to Shojiro Nishio, Information and Computer Sciences Dept., Faculty of Engineering Science, Osaka Univ., Toyonaka, Osaka, 560 Japan, phone 81 (6) 853-5737.

First Int'l Conf. on Deductive and Object-Oriented Databases: Dec. 4-6, 1989, Kyoto, Japan. Cosponsors: Information Processing Society of Japan et al. Submit paper by **May 1, 1989**, to Won Kim, MCC, 3500 W. Balcones Center Dr., Austin, TX 78759, phone (512) 343-0860; Jean-Marie Nicholas, ECRC Arabellastr. 17, 8000 Munich 81, FRG, phone 49 (89) 926-99-110; or Shojiro Nishio, Information and Computer Sciences Dept., Faculty of Engineering Science, Osaka Univ., Toyonaka, Osaka, 560 Japan, phone 81 (6) 853-5737.

Fourth Knowledge Acquisition for Knowledge-Based Systems Workshop: Oct. 1-6, 1989, Banff, Canada. Sponsor: American Assoc. for Artificial Intelligence. Submit draft paper by **May 1, 1989**, John H. Boose, Advanced Technology Center, Boeing Computer Services, 7L-64, PO Box 24346, Seattle, WA 98124, phone (206) 865-3253.

J. Microcomputer Applications: January 1990. A special issue is planned on transputer applications. Submit paper by **May 1**,

1989, to F.J. Rammig, Paderborn Univ., FB 17, Warburger Str. 100, 4790 Paderborn, West Germany, phone 49 (05251) 60-20-69.

Int'l Conf. on Expert Planning Systems: June 27-29, 1989, Brighton, England. Sponsor: Institution of Electrical Engineers. Submit paper by **May 5, 1989**, to Conf. Services, IEE, Savoy Pl., London WC2R 0BL, UK, phone 44 (1) 240-1871, ext. 222.

IEEE Software: March, 1990. A special issue is planned on software development metrics. Submit paper by **May 15, 1989**, to Peter Dyson, Software Productivity Solutions, 122 N. Fourth Ave., Indialantic, FL 32903, phone (407) 984-3370.

14th Conf. on Local Computer Networks: Oct. 10-12, 1989, Minneapolis. Submit paper by **May 15, 1989**, to Larry Green, Protocol Engines, 1421 State St., Santa Barbara, CA 93101, phone (805) 965-0825.

Ninth Conf. on Foundations of Software Technology and Theoretical Computer Science: Dec. 19-21, 1989, Bangalore, India. Submit paper by **May 15, 1989**, to C.E. Veni Madhavan, Tata Research Development and Design Center, 1 Mangaldas Rd., Pune 411001, India, phone (212) 662-453.

CIPS Edmonton 89: Oct. 30-Nov. 1, 1989, Edmonton, Alta., Canada. Sponsor: Canadian Information Processing Society. Submit paper by **May 15, 1989**, to Wayne A. Davis,

Computing Science Dept., Univ. of Alberta, Edmonton, Alta, Canada T6G 2H1, phone (403) 492-3976.

FedCASE 89, Federal CASE Conf.: Oct. 30-Nov. 2, 1989, Gaithersburg, Md. Sponsor: Nat'l Inst. of Standards and Technology. Submit tutorial, workshop, panel session paper and/or proposal by **May 15, 1989**, to Maggie H. Law or Wilma M. Osborne, NIST, Technology Bldg., Gaithersburg, MD 20899.

Fifth Aerospace Computer Security Applications Conf.: Dec. 4-8, 1989, Tucson, Ariz. Submit paper by **May 19, 1989**, to Ronald A. Gove, Booz-Allen and Hamilton, 4330 East-West Hwy., Bethesda, MD 20814, phone (301) 951-2395.

Fourth Int'l Workshop on High-Level Synthesis: Oct. 15-18, 1989, Kennebunkport, Maine. Cosponsor: ACM. Submit extended abstract by **June 16, 1989**, to Raul Camposano, IBM Research Div., T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598, phone (914) 945-3971.

First Int'l Conf. on Systems Integration: Apr. 23-26, 1990, Morristown, N.J. Cosponsors: New Jersey Inst. of Technology et al. Submit paper by **July 25, 1989**, to Peter A. Ng, Computer and Information Science Dept., New Jersey Inst. of Technology, Newark, NJ 07102.

Third Annual Parallel Processing Symposium

Sponsored by the OC IEEE Computer Society, CSUF & ADPA
Location: CSUF, Fullerton, CA
March 29-31, 1989

- Features Keynote Speakers,
- Over 25 Sessions on
- Parallel Processing,
- Neural Networks,
- Neurocomputers.
- Commercial Presentations and Demos.

Over 80 papers submitted from the US and 14 other countries.

See February 89 Computer Magazine
Page 85 For Details.

Call CSUF EE Dept., Fullerton, CA
Dr. Clymer or Dr. Ghazenshahi
(714) 773-3013 for
Information on Registration.

or Contact Dr. Larry Canter for
Commercial Participation c/o
Computer Systems Approach, Inc.
1140 S. Raymond Ave., Suite B
Fullerton, CA 92631
(714) 738-3414



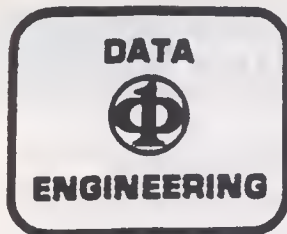
Have you heard about our...

Technical Committee on Robotics?

For information on this, or any of our 32 other TCs, members* may contact

IEEE COMPUTER SOCIETY
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Telephone: (714) 821-8380

Nonmembers: Join us. For more information, circle number 202 on the reader service card.



Sixth International Conference on Data Engineering

February 5-9, 1990

Los Angeles, California, USA

Sponsored by the IEEE Computer Society, Technical Committee on Data Engineering



COMMITTEE

Steering Committee:

C. V. Ramamoorthy (Chair)
University of California, Berkeley

P. Bruce Berra
Syracuse University

Gio Wiederhold
Stanford University

Benjamin W. Wah
University of Illinois

John Carlis
University of Minnesota

General Chairperson:

Joseph E. Urban
University of Miami

Program Chairperson:

Ming T. (Mike) Liu
Ohio State University

Program Vice-Chairpersons:

Hamideh Afsarmanesh
Cal. State Univ., Dominguez Hills

Mohan Ahuja
Ohio State University

Roger King
University of Colorado

Raymond Liuzzi
USAF RADC

Sham Navathe
University of Florida

Amit P. Sheth
UNISYS

Kyu-Young Whang
IBM Research, Yorktown

Industrial Coordinator:

David S. Reiner
Lotus

Asian Coordinator:

Mas Tsuchiya
Syplex International

European Coordinator:

Erich J. Neuhold
GMD, West Germany

Tutorials Chairperson:

Ahmed K. Elmagarmid
Purdue University

Awards:

Tse-yun Feng
Pennsylvania State University

Publicity:

Kane H. Kim
Univ. of California, Irvine

Treasurer:

Wei-Tek Tsai
University of Minnesota

Local Arrangements:

Felipe Carino, Jr.
Teradata

TC on Data Engineering:

Larry Kerschberg
George Mason University

SCOPE

Data Engineering is concerned with the semantics and structuring of data in information systems design, development, management, and use; and with computer system and architectural considerations that are relevant to that concern. It encompasses both traditional and emerging issues and applications. The purpose of this conference is to provide a forum for the sharing of practical experiences and research advances from an engineering point of view among those interested in automated data and knowledge management. Our expectation is that this sharing will enable future information systems to be more efficient and effective, and future research to be more relevant and timely.

We are particularly soliciting industrial, business, and government participation. We know it is vital that there be a dialogue between practitioners and researchers. We look forward to reports of information systems experience detailing experiments, evaluation, problems, and opportunities associated with design, implementation, and operation. Such reports will be given special consideration.

TOPICS OF INTEREST INCLUDE

- AI and Knowledge Based Systems
- Applications and Application Systems
- Benchmarks and Performance Evaluation
- Data Engineering Technique and Tools
- Database Design and Modeling
- Database Management and Structures
- Deductive and Extensive Databases
- Distributed Database Control
- Distributed Database Systems
- Integrity and Security Techniques
- Object-Oriented Database Systems
- Query Languages and Processing

PAPER SUBMISSIONS

Each paper's length should be limited to 8 proceedings pages, which is about 5000 words, or 25 double spaced typed pages. Five copies of completed papers should be mailed before June 15, 1989 to:

Ming T. (Mike) Liu, Department of Computer and Information Science, Ohio State University, 2036 Neil Avenue, Columbus, OH 43210-1277, USA; (614) 292-1837; Liu@CIS.IRCC.Ohio-State.Edu

TUTORIALS

The day preceding the conference will be devoted to introductory tutorials which may provide the background for the conference proper. The day following the conference will be devoted to the advanced tutorials. Proposals for tutorials on data engineering topics are welcome. Send proposals by June 15, 1989 to:

Ahmed K. Elmagarmid, Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA; (317) 494-1998; AKE@cs.purdue.edu

CONFERENCE TIMETABLE AND INFORMATION

Papers Due:	June 15, 1989
Tutorial Proposals Due:	June 15, 1989
Acceptance Letter Sent:	September 15, 1989
Camera Ready Copy Due:	November 1, 1989
Tutorials:	February 5 and 9, 1990
Conference:	February 6-8, 1990

For further information contact the General Chairperson, Joseph E. Urban, College of Engineering, University of Miami, P.O. Box 248294, Coral Gables, FL 33124; (305) 284-2404

AWARDS, STUDENT PAPERS AND SUBSEQUENT PUBLICATIONS

Awards will be given to the best paper and to the best student paper (denoted as such when submitted solely by students). The latter will receive the K. S. Fu award honoring one of the early supporters of the conference. Up to three grants of \$500.00 each will be available to help defray travel costs of student authors. Outstanding papers will be considered for publication in the IEEE Computer Society publications: *Computer*, *IEEE Expert*, *IEEE Software*, *IEEE Transactions on Knowledge and Data Engineering*, etc. For more information, contact the general chairperson.

CAREER OPPORTUNITIES

RATES: \$12.00 per line, \$120 minimum charge (up to ten lines). Average five typeset words per line, eight lines per column inch. Add \$10 for box number. Send copy at least one month prior to publication to: **Heidi Rex or Marian Tibayan, Classified Advertising, COMPUTER Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380.**

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *COMPUTER* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...", "...1-4 years maximum experience...", "...up to 5 years experience...", or "...10 years maximum experience." *COMPUTER* reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience ranges are suggested minimum requirements, not maximums." *COMPUTER* assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.

SANTA CLARA UNIVERSITY **Tenure Track Position In** **Computer Engineering**

Santa Clara University's EECS department has an immediate opening for a tenure track faculty position in Computer Engineering. Candidates with specialization in Computer Architecture or Software Engineering.

Applicants must have a Ph.D. in Computer Engineering or a closely related area with a commitment to both teaching and research. Senior applicants must have a distinguished research and teaching record.

SCU is located 45 miles south of San Francisco, in the center of Silicon Valley. The department offers B.S., M.S., and Ph.D. degrees in both computer and Electrical Engineering. Engineering computing facilities include more than 70 engineering workstations, besides a VAX-11/750. University academic computing resources include a VAX-8650 plus more than 800 IBM PCs. Most computing facilities are networked together over a campus wide local area network.

Candidates should submit a resume including the names of at least three references to:

Dr. Hasan S. Alkhatib, Chairman
Faculty Search Committee

EECS Department

Santa Clara University

Santa Clara, California 95053

E-mail: HALKHATIB@SCU.BITNET

Santa Clara University is an Affirmative Action/Equal Opportunity Employer.

XAVIER UNIVERSITY OF LOUISIANA

Tenure track position to begin in August 1989. Qualifications: Ph.D. or Masters degree in Computer Science from an accredited institution. Expect to teach 3 to 4 undergraduate courses per semester. Xavier University of Louisiana is a Catholic Liberal Arts University serving a predominantly Black student body. The Computer Science Department offers 2 programs leading to a B.A. in Computer Science or Computer Information Systems. Send resume or further inquiries by May 30, 1989 to: Ms. Fatma Dandashi, Acting Chair, Computer Science Department, P.O. Box 50A, Xavier University, New Orleans, LA 70125, or call (504) 483-7458.

UNIVERSITY OF **SOUTHERN CALIFORNIA** **Chair, Computer Science**

The Computer Science Department of the School of Engineering, University of Southern California (USC), seeks a distinguished computer scientist with the vision and skill to lead a strong department and further strengthen it. The School of Engineering, with 170 faculty and 4,300 students, ranks 6th in the nation in sponsored research expenditures. The Computer Science Department has a full time faculty of 22, and a student body of 125 Ph.D. students, 250 M.S. candidates, and 200 undergraduates. The departmental research budget currently is \$3M per year. Faculty research interests encompass the traditional areas of computer science, plus interdisciplinary, emerging fields such as robotics and neural computation. Computing research support is provided by a large network of modern workstations, and various supercomputers are available through the network. More than 100 SUN workstations are used for teaching. The department maintains close ties with the Electrical Engineering Department which has a strong Computer Engineering Group of 16 faculty, and with the Information Sciences Institute (ISI), an off-campus research facility of the School of Engineering. ISI's staff of 200 conducts research on a broad spectrum of computer science topics. Southern California has the highest industrial production in the nation. Opportunities for industry/university collaboration are excellent because much of the local industry is high-tech and computationally oriented.

Please send nominations and applications to:

Professor Aristides Requicha

Chair, Search Committee

Computer Science Department

University of Southern California

Los Angeles, CA 90089-0782

Telephone: (213) 743-3805

Net Mail: requicha@lipari.usc.edu

Applications should include a resume and a list of professional references. USC is an equal opportunity employer.

UNIVERSITY OF CALIFORNIA **AT RIVERSIDE** **Positions in Computer Science**

Applications and nominations are invited for tenured or tenure track positions in Computer Science beginning July 1, 1989 or later. The positions are open as to rank; candidates at all levels, including Full Professor, will be considered. Ph.D. and demonstrated excellence in research and teaching are required. All areas of Computer Science will be considered, including but not restricted to Programming Environment, Computer Systems, Theory of Computation, and Design and Development Automation. Salary and rank are determined by established criteria of the University of California. In addition to supplying a curriculum vita and a list of publications, candidates should either arrange for at least three letters of reference to be sent or submit the names of at least three references. The above materials should be sent to:

Professor Gerhard Gierz

Chair, Computer Science Recruiting Committee

Department of Mathematics and Computer Science

University of California

Riverside, CA 92521

The pool of candidates will consist of all those whose completed applications (consisting of a vita, a list of publications, and at least three letters of reference) are received by February 28, 1989. If the search has not been successfully completed by April 5, 1989, the pool will be enlarged to include all those whose completed applications are received between March 1, 1989 and April 5, 1989.

University of California, Riverside, is an Affirmative Action/Equal Opportunity Employer.

RENSSELAER POLYTECHNIC **INSTITUTE**

The Computer Science Department invites applications for tenure-track faculty positions at all academic ranks. Research, visiting and postdoctoral appointments may also be available. Applicants should have a Ph.D. in Computer Science (or a related area) and a commitment to excellence in teaching and research. Preferred research interests are parallel computation, database systems, computer graphics, computer vision, image processing, VLSI systems and symbolic computation; however, all areas will be considered. The department offers B.S., M.S., and Ph.D. degrees in Computer Science and has excellent computing facilities. Send resumes and at least three references to Joseph E. Flaherty, Chairman, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180-3590. Rensselaer is an Equal Opportunity/Affirmative Action Employer.

UNIVERSITY OF DELAWARE

Department of Computer and Information Sciences

Are you interested in joining the computer science faculty of a growing, dynamic department in an attractive university town within easy traveling distance to New York, Philadelphia, Baltimore, and Washington? The University of Delaware, centrally located on the East Coast, is recruiting for possible openings for tenure-track and visiting faculty positions in the Department of Computer and Information Sciences beginning September 1, 1989. The department has active groups working in artificial intelligence, networks, and symbolic mathematical computation. Special interest exists for candidates with research expertise in architecture, operating systems, parallel processing, databases, theory of computation, programming languages, or software engineering. Strong applicants in all areas of computer science are encouraged to apply.

A Ph.D. degree or its equivalent, and excellence in research and teaching are required. Salary and rank will be commensurate with the candidate's qualifications and experience.

The Department research facilities include various workstations (primarily Symbolics Lisp machines and SUN's) and facilities in a joint research lab shared with the Department of Electrical Engineering. The latter includes a Sequent Symmetry, VAX-8500, three VAX 780's and various other smaller machines. We are well connected to the major networks.

The Computer and Information Sciences Department offers bachelor, master, and doctoral degrees. Resources devoted to academic use in the University Computing Center include an IBM 3090-300 with a vector facility, a CDC Cyber 180, Unix machines (Vax 8650, Pyramid 98xe, Sun 4 with 128 MB memory, 12 Sun 3/60's), and more than 75 microcomputers (IBM PC-XT's, AT's and Macintosh's).

Candidates should send a curriculum vitae and the names of three references to Dr. David Saunders, Acting Chair, Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716. Positions are open until filled.

The University of Delaware is an equal opportunity, affirmative action employer. Applications from members of minority groups and woman are encouraged.

CASE INSTITUTE OF TECHNOLOGY

Case Western Reserve University

We invite applications for tenure track faculty positions at all levels. Candidates from all research areas will be considered, but the thrust research areas in the Department are VLSI systems and design automation, applied artificial intelligence and logic programming, database design and systems, and software systems and design environments, with additional faculty in computer graphics and networks/distributed systems. Candidates should have a Ph.D. in computer science or computer engineering or closely allied fields; competitive salaries will be offered to attract the best candidates.

CWRU is a private university with a total enrollment of 8,400, of which 5,100 are graduate and professional students. The Engineering School of Case Institute of Technology is among the top ten engineering schools in terms of research funding per faculty member and undergraduate student quality. The University campus is the hub of the pleasant area known as University Circle, an incorporation with neighboring cultural centers and museums, about five miles from downtown Cleveland.

The Department of Computer Engineering and Science has a faculty of 13 and growing, and a graduate student body of 140 students, 50 of which are in the Ph.D. program. Departmental facilities are based upon an ethernet local area network, connected to INTERNET, which supports a UNIX operating system and about 30 SUN workstations, color graphics display terminals, together with several printers. In addition, faculty and students participating in the Center for Automation and Intelligent Systems Research have access to the Center's VAX-11/782 and workstations.

The Department recently acquired the Nord Professorship, supported by a donation of one and a half million dollars, for which we invite distinguished senior faculty applicants. This position will provide additional funds for travel, graduate student support and equipment.

Applicants should submit their curriculum vitae and names of at least three references to: Lee J. White, Chairman, Department of Computer Engineering and Science, Case Western Reserve University, Cleveland, Ohio 44106; INTERNET: leew@alpha.ces.cwru.edu; candidates with previous academic experience may wish to provide at most three reprints of their most important publications.

An equal employment and affirmative action employer.

OREGON GRADUATE CENTER

How would you like to work in an academic environment with an active graduate program but no undergraduate teaching responsibilities? A place where research is a serious concern of everyone, with excellent facilities and a minimum of administrative red tape? The Oregon Graduate Center is such a place and we are expanding our program in Computer Science and Engineering. We seek faculty colleagues with experience in graduate education and with strong research interests. Some technical areas of particular importance to us are parallel computation, connectionist architectures, VLSI design and distributed database systems.

OGC is located in Portland, which remains the most affordable of the Pacific Coast's beautiful cities. It offers a place to combine a relaxed life style with an intense work style focused on the things you most want to do.

For more information about us, please send inquiries to: Prof. Richard Kiebertz, Chairman, Dept. of Computer Science and Engineering, Oregon Graduate Center, 19600 NW von Neumann Drive, Beaverton, OR 97006.

OGC is an equal opportunity employer.

SOFTWARE DEVELOPER

Full-time position to work for software consulting firm located in Central Ohio. Duties involve design and development of software for user interface applications. Candidate must have M.S. degree in computer science and one year of experience in position offered or related occupations (software developer or computer graphics specialist). In addition, must have one year of experience in software development of computer graphics or user interfaces. Must have developed two user interface systems for professional applications, each requiring a minimum of 5000 lines of C on UNIX, 2D applications features (lines, curves, bit maps, polygons), a minimum of 10 on-line help features and a minimum of two windows. Must also have developed two user interface enhancement/development tools including window applications. Must have completed one rendering project for raster devices. Specialized experience may be gained concurrently and may be included in experience in position offered or related occupations. Salary: \$45,000 per year. 40 hrs./wk. Send resume to R. Lechler, JO #1167564, Ohio Bureau of Employment Services, P.O. Box 1618, Columbus, Ohio 43216.
AN EQUAL OPPORTUNITY EMPLOYER

UNIVERSITY OF CENTRAL FLORIDA

Computer Engineering

The University of Central Florida College of Engineering invites applicants for a tenure-track Assistant Professor position in its Department of Computer Engineering. A Ph.D. in Computer Engineering or a related discipline is required by the time of employment, and the preferred specialty area is digital systems/computer architecture. UCF is a member of the Florida State University System and has a current enrollment of approximately 18,000 students. The University is developing a research park adjacent to its main campus to support high-technology government and industrial activity in the Central Florida area. Computer facilities for instruction and research include an IBM 4381, a VAX with 8 MB memory, a Gould 32/6780, 2 IBM 5080 color workstations, and numerous mini and micro computers. The Department operates an Engineering AI Laboratory with 9 Symbolics Workstations and a Texas Instruments Explorer System. An Ethernet LAN using TCP/IP protocols links College of Engineering offices and labs together. Each faculty office has a private personal computer with a LAN connection.

Send curriculum vitae and the names of three references postmarked by March 30, 1989 to:

Dr. C.S. Bauer, Chair
Department of Computer Engineering
University of Central Florida
Orlando, Florida 32816
Phone: (407) 275-2236

The University of Central Florida is an Equal Opportunity /Affirmative Action employer. As an agency of the State of Florida, the University makes all application materials and selection procedures available for public review.

**RENSSELAER POLYTECHNIC
INSTITUTE**
Faculty Positions
**Department of Electrical, Computer,
and Systems Engineering**

Applications are invited for tenure-track faculty positions at all levels. Specific areas of interest include: (1) computer engineering, architecture, and performance evaluation, particularly related to parallel image processing algorithms and architectures; (2) computer and communications networks, architectures and protocols; (3) solid-state and optoelectronic devices, interconnects, and systems. The ECSE Department is the largest academic unit at Rensselaer and has a rich tradition of research and education. The department is seeking to add faculty who bring an innovative approach to research and teaching. Active programs in computer engineering, solid-state electronics and integrated circuit design, control systems, robotics and automation, information and decision systems, communications and signal processing, electronics and circuits, and fusion plasma systems contribute to a dynamic research environment. In addition to the extensive research facilities of the department, there are opportunities to initiate or participate in interdisciplinary research programs in one of the major research centers of the School of Engineering, including the Center for Integrated Electronics, the Center for Rensselaer Design Research, the Center for Manufacturing Productivity and Technology Transfer, and the NASA Center for Intelligent Robotic Systems in Space Exploration. New faculty are eligible for special arrangements including summer support, equipment, graduate student support, and reduced teaching load in order to encourage growth of their research programs. Applications or requests for more information should be directed to:

Dr. Arthur C. Sanderson, Chairman
Electrical, Computer and Systems
Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

RPI is an affirmative action/equal opportunity employer.

Rochester Institute of Technology
School of Computer Science
**Department of Applied Computer
Studies**
Software Engineering Position

Visiting or Tenure Track faculty position available for individual who has experience in the instruction of Software Engineering topics, both managerial and technical. Ph.D. or Master's (with professional experience) degree in appropriate discipline required. Duties will include teaching at the graduate level (M.S.) as well as applied research in Software Engineering.

Send resume to Prof. Guy Johnson, Department of Applied Computer Studies, 1 Lomb Memorial Drive, RIT, Rochester, New York, 14623. RIT is an equal opportunity affirmative action employer.

CARNEGIE MELLON UNIVERSITY
**Faculty Position in Field Robotics/
Construction Automation/Sensing**

Carnegie Mellon University, Department of Civil Engineering, invites applications for a tenure-track position as an Assistant Professor to begin in September 1989. Candidates for this position should have an earned doctorate and research experience in field robotics, remote sensing or construction automation, and must be able to interact with other faculty working in robotics, computer-aided engineering and construction project management. Candidates should have strong analytical capabilities (preferably in mechanics or control) and/or experience with microprocessor-based systems, sensors and instrumentation. While a background in engineering is necessary, training in civil engineering is not a requirement. Responsibilities include teaching undergraduate and graduate courses, supervising graduate students and developing a funded research program. Current field robotics/construction automation activities (including fundamental studies in mobility, vision, force sensing, and system integration) in the department are well established, and many opportunities exist for professional and intellectual growth. The department also has close ties with the university's Robotics Institute, and a joint appointment with the Robotics Institute is possible. Interested persons should contact Professor Chris Hendrickson, Faculty Search Committee, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890. Carnegie Mellon is an equal opportunity/affirmative action employer.

**UNIVERSITY OF CALIFORNIA,
LOS ANGELES**
COMPUTER SCIENCE DEPARTMENT

The Department of Computer Science at the University of California, Los Angeles, invites applications for tenure-track positions at the Assistant Professor level in Computer Science beginning in July 1989. Applicants should possess the Ph.D. in Computer Science by July 1989. Applications are also welcome from highly distinguished candidates at the senior level.

Quality is our key criterion for selecting applicants. We expect them to have a strong commitment to both research and teaching and an outstanding record of research for their level. It is important that they exhibit strong potential for continued excellence in university research.

We seek applicants in any mainstream area of Computer Science and we particularly welcome those with research strength in software related areas.

Interested applicants should send a letter of application, a resume, and the names of four references to:

Professor Wesley W. Chu, Chair
Computer Science Department
Attn: Ms. June Myers
Boelter Hall 3732
University of California, Los Angeles
Los Angeles, CA 90024-1596

The University of California is an Affirmative Action/Equal Opportunity Employer.

ROCKFORD COLLEGE
Rockford, Illinois
Computer Science/Mathematics

Rockford College, a small liberal arts college in northern Illinois, invites applications for a tenure-track position in computer science/mathematics. Responsibilities will include teaching approximately twelve hours a week per semester, student advising, and committee work.

Qualified applicants must have a master's degree with post-graduate level work in computer science and mathematics (a Ph.D. is preferred), be qualified to teach upper division computer science and post-calculus mathematics, and present excellent teaching recommendations.

The computer science and math departments collectively consist of six faculty using an IBM 4381 VM, operating under CMS with 30 terminals and a lab of 16 IBM PC's and XT's. Curriculum is ACM-based.

Rank and salary are commensurate with qualifications, and there is an attractive benefits program. Send resume and three letters of teaching recommendations to Dr. Gerald Caton, Computer Science Department Chair, Rockford College, 5050 E. State Street, Rockford, Illinois 61108-2393. AA/EOE

ITHACA COLLEGE
**Department of Mathematics &
Computer Science**

The Department of Mathematics and Computer Science has a tenure eligible position for Fall 1989 in computer science at the rank of assistant professor. A Ph.D. in computer science or a related discipline with the ability to teach a variety of computer science courses is required. Send vitae to Dr. Patricia Woodworth, Department of Mathematics and Computer Science, Ithaca College, Ithaca, New York 14850. An Equal Opportunity/Affirmative Action employer.

UNIVERSITY OF WASHINGTON

The University of Washington seeks candidates for tenure-track faculty appointments in computer science/engineering starting in the 1989-90 academic year. Applications from outstanding individuals in all areas of computer science/engineering will be considered.

A moderate teaching load allows time for quality research and close involvement with students. We expect applicants to have a strong commitment to both research and teaching, and an outstanding record of research for their level. Any appointment should bring significant new research strength to the University.

Interested applicants should send a letter of application, a resume, and the names of four references to: Computer Science/Engineering Faculty Recruiting Committee, FR-35, University of Washington, Seattle, WA 98195.

The University of Washington is an Affirmative Action/Equal Opportunity Employer. The Ph.D. is required for these positions.

FLORIDA ATLANTIC UNIVERSITY **Computer Engineering**

The Department of Computer Engineering has tenure track openings starting in August, 1989. The department has developed from a group within the Electrical and Computer Engineering Department. It has an active graduate program with approximately 100 master's students and 25 Ph.D. students and expects to add a bachelor's program during the coming year. It cooperates closely with the Department of Computer Science, also housed in the College of Engineering.

Candidates should have an earned doctorate in a relevant field and should be interested in both teaching and developing a strong sponsored research program.

Interests of the department center around computer networks and communications, architecture, including parallel computers, VLSI, fault tolerant and secure systems, artificial intelligence, and microprocessor based systems.

Florida Atlantic University, part of the State University system, is located on the southeast coast of Florida, close to many computer and electronics companies, such as IBM, Motorola, Siemens and ModComp.

Please send a complete resume, including the names of at least three references to Dr. Alan B. Marcovitz, Acting Chairman, Department of Computer Engineering, Florida Atlantic University, Boca Raton, FL 33431 no later than April 15, 1989.

Florida Atlantic University is an equal opportunity/affirmative action employer.

STANFORD UNIVERSITY

The Department of Electrical Engineering of Stanford University is seeking candidates for a tenured or tenure-track position in *Fiber-Optic Telecommunications*. The field of interest is interpreted broadly to include both systems and device aspects of optical communications. However, the individual we seek should have greatest strength and experience with systems, and in particular should be knowledgeable of the commercial telecommunications industry and its needs. Candidates should have excellent knowledge of the latest theoretical results and a proven skill in implementing such knowledge in "proof of concept" demonstrations. Simultaneously, he or she should have a good knowledge of the relevant device physics and technology. The appointment can be made at either a junior or a senior level, but the Department prefers a seasoned individual, possibly with industrial experience, and having the potential to lead a larger telecommunications effort at Stanford. Applicants should have an earned Ph.D., demonstrated research ability, and a strong interest in graduate and undergraduate teaching. Stanford University is an Equal Opportunity Employer and welcomes applications from women and members of minority groups. Please submit, no later than May 15, 1989, a detailed resume, a publication list, and the names of five references to Professor Allen Peterson, Chairman, Search Committee, Durand 227, Department of Electrical Engineering, Stanford University, Stanford, California 94305-4055.

CALIFORNIA STATE UNIVERSITY, FRESNO **Faculty Positions in Computer Science**

Candidates for appointment as Assistant or Associate Professor should have a Ph.D. in Computer Science or equivalent; show evidence of successful teaching; and have a research record which demonstrates the interest and ability to develop a strong research effort at CSU Fresno. Specialty areas of particular interest to the department include computer architecture, parallel and distributed computing, high-level language design, and graphics.

Candidates for appointment as Lecturer should have an MS in Computer Science or closely-related field and demonstrate successful teaching experience in computer science. Applications for Visiting Faculty positions are welcome.

The Computer Science Department offers a BS in Computer Science and has proposed an MS degree. Department computing facilities include a VAX 11/785, an AT&T 3B15, a networked Sun Microsystems lab, and a Convergent Technology micro lab. The CSUF Computer Services operates a range of computers and microcomputer labs available for use by both faculty and students. In addition, via our INTERNET connection with CSU system facilities, we have access to a variety of parallel and supercomputers.

Candidates should submit a resume, transcripts, and a statement of professional interests and goals and arrange for three letters of reference. Nominations are encouraged. Address all correspondence and confidential papers to: Dr. Brandt Kehoe, Chair, Department of Computer Science, CSU Fresno, Fresno, CA 93740-0109. Phone: (209) 294-4373; E-mail: kehoe@CSUFresno.edu

To insure full consideration, applicants are encouraged to have all application material on file by March 31, 1989.

An Affirmative Action/Equal Employment Opportunity Employer.

JOHANNES KEPLER UNIVERSITY IN LINZ (AUSTRIA) **(Research Institute for Symbolic Computation)** **Tenured Position of a Full Professor for Symbolic Computation**

Applications are invited for a newly created tenured position of a full professor at the School of Technical Sciences, Johannes Kepler University in Linz, Austria. Candidates should have an outstanding research record in at least one of the main areas of symbolic computation (computer algebra, computer analysis, computational geometry, computational logic, automatic programming etc.), a commitment to excellence in graduate teaching, and the ability and willingness to combine, in their field of expertise, mathematical/algebraic foundations with potential industrial applications (for example, expert systems, geometrical modeling, robot programming, scientific software). Teaching and supervising of students may also be in English. Candidates from foreign countries are explicitly encouraged.

The School of Technical Sciences has installed an independent institute for symbolic computation (RISC-LINZ, Research Institute for Symbolic Computation) with an 8 member faculty. RISC-LINZ operates in close interaction with the Department of Computer Science (a twenty-six member faculty) and the Department of Mathematics (a twenty-two member faculty).

At Johannes Kepler University, the symbolic computation effort is a part of and a major driving force for a planned expansion (by 10 full professorships) of the School of Technical Science in the direction of "mechatronics" (intelligent control of technical production) in close cooperation with Austrian industry.

Applicants should send a resume (together with a few important publications representing the candidates research expertise) to the Dean of the School of Technical Sciences, Prof. Dr. Peter Weib, Johannes Kepler University, A4040 Linz (Austria). (Telephone Austria (732) 2468-312.), by March 31, 1989. Further information can also be obtained from the Chairman of RISC-LINZ, Prof. Dr. Bruno Buchberger, Johannes Kepler University, A4040 Linz. (Tel: Austria (732) 2468-9219. Electronic mail: K313370@AEARN.bitnet.)

TULANE

Tulane is a major private university, selective in enrollment and comprehensive in scope, with about 10,000 undergraduate and graduate students. The campus is located in a residential area of uptown New Orleans, one of America's most distinctive cities. Faculty benefits include dependent tuition, mortgage assistance, relocation expenses, 12% TIAA and typical insurance benefits.

The department computer network includes a Pyramid 9810 running UNIX, a VAX 11/780 running VMS, and several Sun workstations. This local area network is a subnet of the Tulane University network which is connected to SURANET, a regional subnet of NSFNET. Other major computing resources on the Tulane University network and accessible from Computer Science Offices include an IBM 3081KX mainframe. In addition, an IBM 4341 supports expert system research and instruction.

The department invites applications for faculty positions at all levels for the spring, 1989. Candidates should have a Ph.D. in Computer Science or a related area. Research excellence or demonstrated research potential and a commitment to quality teaching are required. Applicants for instructorships must possess a masters in Computer Science or ABD.

Applications should be directed to Dr. Johnette Hassell, Head, Department of Computer Science, School of Engineering, Tulane University, New Orleans, LA 70118. Applications must be received by April 16, 1989 for fall appointment. Tulane University is an Affirmative Action Equal Opportunity Employer.

HARVEY MUDD COLLEGE
Position Announcement in Computer
Science

Applications are invited for a one-year visiting position in Computer Science beginning July 1, 1989. An applicant should have, or be close to completion of, a Ph.D. in Computer Science and have a strong interest in both undergraduate teaching and research. Candidates from all areas of computer science or a computer-related field will be considered. Responsibilities include teaching, research, and the supervision of industrially-sponsored projects in computer science. The Computer Science Department's facilities include a network composed of a VAX 11/750 and a Sequent Symmetry. The Computer Science network is also interfaced with the college network of VMS machines: a Micro VAX 2, 4 VAX 11/750s, and a VAX 8600.

Harvey Mudd College, one of the nation's most selective undergraduate colleges, is a member of The Claremont Colleges. The College is an equal opportunity/affirmative action employer. Please send resume and the names of four references to Professor Wing C. Tam, Computer Science Department, Harvey Mudd College, Claremont, CA 91711.

FLORIDA ATLANTIC UNIVERSITY
Chairman, Computer Engineering

The Department of Computer Engineering at FAU is searching for a chairman who can provide intellectual and administrative leadership. Candidates should have an earned doctorate in a relevant field and a distinguished record in research and teaching.

The Department has recently developed from a group within the Electrical and Computer Engineering Department. It has an active research and graduate program with approximately 100 master's students and 25 Ph.D. students and expects to add a bachelor's program during the coming year. It cooperates closely with the Department of Computer Science, also housed in the College of Engineering. Both departments will be located in the new Science/Engineering building, now under construction.

Interests of the department center around computer networks and communications, architecture, including parallel computers, VLSI, fault tolerant and secure systems, artificial intelligence, and microprocessor based systems.

Florida Atlantic University, part of the State University system, is located on the southeast coast of Florida, close to many computer and electronics companies, such as IBM, Motorola, Siemens and ModComp. The Department maintains close ties with these and other high tech companies.

Please send a complete resume, including the names of at least three references to Dr. Eduardo B. Fernandez, Department of Computer Engineering, Florida Atlantic University, Boca Raton, FL 33431 no later than April 15, 1989.

Florida Atlantic University is an equal opportunity/affirmative action employer.

UNIVERSITY OF WISCONSIN-
MADISON
Faculty Positions

The Department of Electrical and Computer Engineering invites applications for tenure and tenure-track positions. A Ph.D. degree is required, and successful candidates are expected to participate in both teaching and research activities. Applicants in all areas of computer engineering are invited to apply, but the following areas are of special interest: Computer architecture, computer networks, VLSI and computer-aided design, microprocessor and minicomputer applications, real-time control and instrumentation applications, and engineering applications of artificial intelligence. Rank and salary will be commensurate with qualifications and experience. Send resume and names of three references to J. Leon Shohet, Chairman, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, 1415 Johnson Drive, Madison, WI 53706, an equal opportunity/affirmative action employer.

TRINITY COLLEGE
Computer Science

Pending approval by the Board of Trustees, applications are invited for a tenure track position in Computer Science in the Department of Engineering and Computer Science at Trinity College starting in January 1990. The position involves undergraduate and graduate instruction and research. A doctoral degree in Computer Science or equivalent is required. Applicants will be considered in all areas of computer science that complement or strengthen our current curricular/research programs. We are interested in receiving applications from qualified women and minorities. Trinity College is an equal opportunity/affirmative action employer. Please send resume to Professor Joseph D. Bronzino, Chairman, ECS Department, Trinity College, Hartford, CT 06106. Consideration of applications will begin on May 1, 1989 and the search will remain open until the position is filled.

CENTRAL CONNECTICUT STATE
UNIVERSITY

COMPUTER SCIENCE—Fall 1989—One Assistant or Associate Professor and one Associate or Full Professor, tenure track, rank and salary dependent on experience and qualifications. To teach Computer Science at all levels and to establish an active program in some specialty in the discipline. Ph.D. required. Previous teaching experience preferred. Send letter of application with resume and names, addresses and telephone numbers of three references to Clifford L. Pelletier, Department of Mathematics and Computer Science, CENTRAL CONNECTICUT STATE UNIVERSITY, New Britain, CT 06050-4010 by March 1, 1989. CCSU is an AA/EO employer. Women, minorities, handicapped and veterans are encouraged to apply.

THE GEORGE WASHINGTON
UNIVERSITY
Computer Science Faculty Positions

Computer Science faculty positions at the Assistant, Associate and Full Professor ranks are available commencing Fall Semester 1989.

Especially sought are applicants able to conduct research and teach in the areas of image processing, machine vision, artificial intelligence and interactive computer graphics. Candidates should have an earned Doctorate and research experience with an interest in both teaching and research. Ability to attract funded research is valued. The George Washington University is located in the center of Washington, DC. This metropolitan area sustains the second largest concentration of research and development activity in the United States, creating a continuing demand for rigorously-trained engineers and many research opportunities. Our Department is the largest in the School of Engineering & Applied Science with 38 full-time faculty, accredited electrical engineering, computer engineering, and computer science degree programs, a large graduate and undergraduate student body, and a substantial research budget. Current funded research projects include computer graphics, computer security, data transmission standards and compression, fast packet switching, image processing, intelligent user interfaces, laser shields, MHD plants, magnetic devices, medical imaging, multipath fading and encryption, remote sensing, space-based radar, and special-purpose VLSI designs. Send curriculum vita, list of publications and 3 references to:

Professor James D. Foley, Chairman
Department of Electrical Engineering
& Computer Science

School of Engineering & Applied Science
The George Washington University
Washington, DC 20052

The George Washington University is an equal opportunity/affirmative action employer.

UNIVERSITY COLLEGE,
CORK IRELAND
Professorship No. 2 in
Computer Science

Applications are invited for a second Chair in Computer Science which is to be filled from 1 October 1989. Applicants should have demonstrated a high level of research and professional accomplishments in any main stream area of Information Processing.

University College Cork is a constituent college of the National University of Ireland, the largest university in Ireland. It offers undergraduate degree programmes in Computer Science at pass and honours levels and has extensive postgraduate activity. The present staff complement is 8 full-time academics and some part-time lecturers. Laboratory facilities include a range of up-to-date facilities.

Further particulars available from Mr. M.F. Kelleher, Secretary, University College Cork, Ireland, to whom applications should be sent by 31 March 1989.

Tel. +353-21-276871 ext. 2364.

SYSTEMS ENGINEER

Systems Engineer for firm in SW Ohio to provide specialized consultancy on location to design, develop and analyze systems applications software in data-base for UNISYS and IBM mainframe computers using COBOL, C, FORTRAN, LISP, GEMCOS, ALGOL and SAS languages, using interactive conversion techniques, compiler construction; data communication interface and network architecture. Require B.S. in Computer, Information Systems, or Management Engineering with 2 years' experience in job described or M.S. in any of these disciplines and 2 years' of described experience. Extensive travel (75%) to various locations (Indianapolis, Columbus, Cleveland, Cincinnati) for periods of 3-9 mo. per project/assignment (expenses reimbursed). 40 hrs/wk, 8:30am-5pm. \$640/wk. Qualified applicants reply immediately with resume to R. Lechler, JO#1170804, Ohio Bureau of Employment Services, P.O. Box 1618, Columbus, OH 43216.

DEPARTMENT HEAD— COMPUTER SCIENCE University of Lowell Lowell, Massachusetts

The faculty of Computer Science is seeking a new Department Head to provide leadership in research and to administer a comprehensive Graduate and Undergraduate program. This search will continue during the 1988-89 academic year or until the position is filled. The Department Head will be responsible for faculty recruitment and evaluation, development of a comprehensive research plan and the management of all departmental resources. A major priority is to recruit new faculty and expand research activity and the graduate program in Computer Science.

The successful candidate must hold an earned doctorate in Computer Science or a directly related field and must be a citizen or permanent resident of the U.S. The candidate should be nationally recognized and currently active in research and teaching in at least one area of modern Computer Science. Prior experience in academic and research administration sufficient to qualify for a tenured position at the rank of full professor is also required.

For more information about the CS Department, see our prior ad in the January issue of IEEE Computer (p. 76), ACM Communications or the Affirmative Action Register. Interested applicants are invited to forward a complete academic vitae and names of three references (two in his/her research area), and a letter stating his or her approach toward expanding research and graduate study in Computer Science at the University of Lowell.

Address replies to:

Professor Robert Lechner
Department Head Search Committee
Computer Science, WL229
University of Lowell
Lowell, MA 01854

The University of Lowell is an Equal Opportunity/Affirmative Action, Title IX, 504 Employer.

COLGATE UNIVERSITY Hamilton, New York 13346 Department of Computer Science

We invite applications for a half-time leave replacement position for 1989-1990. This would be an ideal position for someone who wants to combine some teaching with research during a sabbatical. The position may be filled by someone teaching half-time for the year or full-time for one semester (two courses with labs or three without).

Colgate is a quality liberal arts college with a first-rate computer science program. The department has five faculty with the following research interests: theory of computation and programming language semantics, parallel implementations of functional programming languages, temporal reasoning in natural language processing, graphics and chaos, and discrete event simulation on parallel computers. The Computer Science Department has an introductory lab equipped with sixteen PC and PC/AT clones, and an upper-level/research lab with the following equipment: a VAX 750 running BSD 4.3 Unix, a TI Explorer, four 17-node transputer-based parallel computers (which can be configured as a single 68-node network), five Amiga PC's, and PC/AT clones in every faculty office. Over the summer the department will install a network of either Sun or NeXT workstations. The faculty offices and laboratory machines are connected on an ethernet. We are members of both CSNet and BitNet.

Applicants should send a resume and the names of three references to: Chris Nevison, Chairman, Department of Computer Science, Colgate University, Hamilton, NY 13346. All applications received before April 1, 1989 and after that until the position is filled, will be considered. Colgate University is an affirmative action/equal opportunity employer.

WEST VIRGINIA UNIVERSITY Assistant Vice President Computing and Information Resources

West Virginia University invites applications and nominations for the position of Assistant Vice President for Computing and Information Resources. This land grant University has more than 18,000 students enrolled in 175 degree programs and is the state's major research university. The annual budget exceeds 230 million dollars.

RESPONSIBILITIES: The Assistant Vice President reports to the Provost and is responsible for administering academic service personnel and budgets, coordinating support services; recommending the allocation of resources; developing plans and recommending policy for academic and administrative computing, communications, and information resource management; and acting as chief liaison with statewide and national computing resource centers.

QUALIFICATIONS: Candidates for this position will be expected to possess an earned doctorate or masters degree; have held senior level administrative positions; demonstrate a proven record of accomplishment in planning, implementation and management

of computing and telecommunication systems in a multifaceted environment; exhibit an understanding of academic, administrative and research computing needs; possess knowledge of computing, communication and information systems and exhibit the ability to communicate with and provide leadership for a broad range of constituencies.

SALARY: Competitive.

STARTING DATE: July 1, 1989 or as soon as possible after the conclusion of the selection process.

DEADLINE FOR APPLICATIONS: Accepted until the position is filled with screening beginning January 15, 1989.

APPLICATION PROCEDURE: West Virginia University is committed to affirmative action and invites nominations for and applications from women and minority candidates. Submit letter of application, curriculum vita and names, addresses and telephone numbers of three professional references to:

Dr. William E. Vehse, Chairperson
Search Committee for Assistant VP,
Computing & Information Resources
308 Stewart Hall
West Virginia University
Morgantown, WV 26506

An Affirmative Action/Equal Opportunity Employer.

NAVAL POSTGRADUATE SCHOOL Computer Science

The Computer Science Department invites applications for faculty positions at all levels. Our primary interests are in the areas of operating systems and programming languages. Our secondary interests are in the areas of visual data processing, graphics, and computer architecture (especially real-time and parallel-processing aspects of the three). Applicants should have a Ph.D. in Computer Science or a closely related field and be committed to high-quality teaching and research. Senior applicants must have distinguished research records. Appointments can begin at any time.

The Department offers M.S. and Ph.D. degrees in computer science, but no undergraduate degrees. Currently, 110 students are enrolled in the M.S. program and five in the Ph.D. program. Students are military officers or civilian employees of the Department of Defense and are fully supported by their sponsoring organization during their studies. Departmental facilities (supported by eight full-time computer professionals) include six instructional and research laboratories with extensive state-of-the-art equipment. Faculty normally teach for two quarters and perform research for two quarters per year. The Monterey-Carmel area provides a pleasant coastal climate and easy access to Silicon Valley companies.

Send a detailed resume, an abstract of your best recent research, and letters of reference to Faculty Search Committee, Computer Science, Code 52, Naval Postgraduate School, Monterey, CA 93943 (phone (408) 646-2449). An Equal Opportunity/Affirmative Action Employer.

SOFTWARE DEVELOPER

Full-time position to work in Columbus, Ohio, for a software consulting firm. Work schedule: 7:30 a.m. to 4:30 p.m., Monday through Friday. Qualified applicant will be involved in design and development of software for levels 3 and 4 of Signaling System 7 (SS7). Must have a B.S. in computer science or a B.S. in mathematics or electrical engineering or electronic engineering with a specialization in computer science and three years of experience in position offered or related occupation, design engineer. Must also have one year experience or training in design/development of SS7 levels 3 and 4 and one year experience or training in design/development on real time distributed switching operating system. Specialized experience may be included in experience in position offered or related occupation. Salary: \$41,000/Yr. Send resume to R. Lechler, Ohio Bureau of Employment Services, P.O. Box 1618, Columbus, Ohio 43216, Attention Job Order No. 1167502. ANEQUAL OPPORTUNITY EMPLOYER

UNIVERSITY OF KANSAS Electrical and Computer Engineering

The Electrical and Computer Engineering Department of the University of Kansas seeks to fill several tenure-track and visiting faculty openings, available beginning August 1989, or as negotiated. Applicants should have teaching and research interests in Arti-

ficial Intelligence, Software Engineering, or Computer Architectures. Applicants with added experience and/or research record in VLSI will be given first preference. New faculty will be expected to perform teaching and curriculum development activities, and to participate in on-going research activities as well as to develop their own research. Rank and salary will be commensurate with qualifications and experience. A doctorate is required. Resumes should be sent to Dr. James R. Rowland, Chairman, Electrical and Computer Engineering Department, University of Kansas, Lawrence, Kansas 66045-2228. Phone: (913) 864-4620. E-mail: jrowland@volta.ece.ukans.edu. The University of Kansas is an equal opportunity, affirmative action employer.

ACADEMIA SINICA Taiwan, Republic of China Institute of Information Science

Applications are invited for research positions in Institute of Information Science, Academia Sinica. Ph.D. in computer related fields required. Demonstratable research ability necessary. Applicants for senior positions must have proven research record. All fields in computer science are welcome.

The Institute offers a good research environment. No duty of teaching. Facilities include 3 VAX's, 4 SUN's (including a SUN-4/260), 2 E&S workstations, a lot of small computers and an easily accessible ETA-10Q super computer at Academia Sinica. Two-

year expansion plan will strengthen our facilities with an IRIS and a multi-processor machine.

Interested people please send application to Dr. Y. S. Kuo, Acting Director, Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China. e-mail address: WT6B0001@TWNMOE10.BITNET. FAX: (011-886-2) 782-4814.

UNIVERSITY OF GEORGIA

The Computer Science Department has at least one tenure-track faculty position open at the rank of Assistant Professor, effective September, 1989.

Applicants must have a Ph.D. in Computer Science or a closely related discipline, and a strong interest in both teaching and research. The Department seeks applicants to strengthen its programs in all areas of computer science.

Departmental research facilities are hosted by SUN dataverses connected to an Intel IPSC/2 Hypercube, and offer access to extensive University facilities including an ETA-10 Q216 supercomputer.

Applications must be postmarked by March 31, 1989. Send a curriculum vitae and the names of four references to R. W. Robinson, Head, Computer Science Department, 415 Graduate Studies Research Center, University of Georgia, Athens, Georgia 30602.

The University of Georgia is an equal opportunity/affirmative action institution.

Faculty Positions Computer and Information Science

New Jersey Institute of Technology seeks assistant, associate and full professors for spring/fall 1989. Ph.D. in computer science or closely related field required. Senior level applicants must have proven research and funding record. Positions available in, but not limited to: distributed computing including computer architecture, data communications networking, realtime computing and fault tolerance; software development including artificial intelligence, expert systems, computer graphics, office automation, data management systems, cognitive science, and computational linguistics.

Department offers B.S., B.A., M.S., and Ph.D., in computer science, and Ph.D. in management jointly with Rutgers-Newark. Computing facilities include VAX 8800, VAX 8530, IBM 4361, SUN workstations, Symbolics machines, TI Explorers and graphics systems.

NJIT is the comprehensive technological university of New Jersey with nearly 8,000 students enrolled in Newark College of Engineering, the School of Architecture, College of Sciences and Liberal Arts and the School of Industrial Management.

NJIT does not discriminate on the basis of sex, race, color, handicap, national or ethnic origin, or age in employment.

Send resume and names of at least three references to:
Personnel Box CIS



**New Jersey
Institute of Technology**
Newark, New Jersey 07102

COMPUTER/COMMUNICATIONS ENGINEERING FACULTY POSITION

The Department of Electrical, Computer and Systems Engineering at Boston University seeks applications for a faculty position at the Associate/Full-Professor level in the area of Computer/Communications Engineering. The position is in the area of computer architecture and computer communications. Candidates must have an earned doctorate and three or more years' experience in the area of computer architecture and/or computer communications. The successful candidate should have a good record of funded research and refereed publications.

Boston University is located in the heart of Boston's university complex, and provides excellent opportunities for interaction with colleagues from the surrounding universities and the Rts. 128 and 495 technology complex. The department has 33 faculty, 800 undergraduates, and 250 graduate students. There are excellent computer facilities, including VAX's, an IBM 3090, and a Connection Machine.

Applicants should send their curriculum vitae to Professor Thomas G. Kincaid, Chairman, Department of Electrical, Computer and Systems Engineering, Boston University College of Engineering, Boston, MA 02215. Potential applicants who wish to know more about the position before applying should call Professor Kincaid at 617/353-2806.

**Boston University
College of Engineering**

An equal opportunity, affirmative action employer



BOOK REVIEWS

Editor: Wiley McKinzie, School of Computer Science and Technology, Rochester Institute of Technology, Rochester, NY 14623; Compmail, w.mckinzie; CSnet, wrm@rit

Software Evolution: The Software Maintenance Challenge

Lowell Jay Arthur (John Wiley & Sons, New York, 1988, 254 pp., \$36.95)

Software Evolution addresses software maintenance. Its structure follows a master flow chart of the "software evolution process" containing the following activities: change management, impact analysis, planning a system release, code, testing, and system release.

Each chapter elaborates on one of these activities. Two additional chapters address the implementation of software evolution and the management of software maintenance. Each chapter begins with an outline and concludes with a summary.

A "system release" can be corrective, adaptive, or perfective. Corrective maintenance focuses on fixing defects. Adaptive maintenance includes all work related to enhancing the functionality of the software, hardware, and documentation. Perfective maintenance includes all efforts to improve the quality of the software (e.g., improving efficiency or maintainability). In a real-world case, a system release is often made up of two of the three types. For example, while the programmer adds a requested enhancement, he or she spends a little extra time rewriting other portions of the code in the immediate area to make it easier to modify in the future or to make it conform more closely to company standards.

It took me a while to adjust to the author's writing style. Arthur uses short, direct sentences. Upon reflection, I realized more technical books should be written this way. (An author writes a book because he or she has something to say; therefore, he or she should just say it.)

The book and its main example, which is described and expanded on through several chapters, are data processing-oriented. However, non-DP software developers and maintainers can still gain from the book, since its principles are universal (for example, the importance of configuration control and the importance of planned releases).

The author misuses "i.e." and "e.g." numerous times. Either the technical writer who reviewed the draft should be replaced, or the author did not follow his own advice to "use technical writers to

develop documentation" (p. 231).

The book also contains a few small textual and graphical errors, plus grammar switches that gave me the feeling that portions of the book were cut-and-pasted from other documents. Obviously we all do that, but the trick to make it work is to adjust the text 100 percent, not 99 percent.

The book has a bibliography, and references follow a few sentences in the text, but I would have liked more notes directing the reader to specific references for further information on a topic. For example, a statement that the mainte-

Every member of a software maintenance team should receive a copy of this book.

nance effort of a section of code is cut in half because action X has been taken needs to be backed up with a reference to a book or article on software metrics.

One of the book's strong points is its use of questions at the end of every chapter. These questions force the reader to reflect upon the points just discussed. For example, we all should prioritize maintenance work according to certain criteria (difficulty to implement, product safety, customer satisfaction, etc.), but how many of us have prioritized tasks on our current project? The questions stimulate the reader to either start thinking or quickly skip to the next chapter and not face the fact that thinking and planning real-world projects constrained by time, money, politics, etc. is hard work.

I liked the checklist form of presentation used throughout the book, especially in the last two chapters on implementation and management of software maintenance. The Software Engineering Institute (SEI) affiliated with Carnegie Mellon University advocates realistic evaluation of software development and maintenance — namely, asking questions to learn whether processes are in place to

do a good job. Processes are activities such as configuration management, documentation standards and procedures, and design and code walkthroughs that support the development of software. *Software Evolution* — because it is written in a direct, "do this next" style — is in concert with SEI's goal of insuring that each software project has all the processes in place to assure a reasonable chance of success. One of SEI's immediate tasks is to gather data and measure the maturity levels of different software teams. The author discusses the importance of this in his chapter on implementing software evolution.

Every member of a software maintenance team should receive a copy of this book. Maintenance programmers want to do a good job, but they need to operate within a structure. This book offers that structure. The boss can now demand configuration management, and the workers will know what is needed. He or she can plan phased (as opposed to helter-skelter) releases, and the workers will understand the reasoning behind that decision. Everyone on the team will follow the "build a little, test a little" philosophy because they are all working from the same guideline.

It also would not be a bad idea to force developers to read the book and make them appreciate what problems they cause maintainers by writing spaghetti code, not using parameterized subroutines, not using meaningful names, etc. In fact, one of the author's recommendations is to rotate development programmers through a maintenance group to make them aware of how much the design and implementation decisions made during development influence the ease and quality of maintenance for years to come.

Software Evolution addresses the critical task of software maintenance. The author classifies and discusses each type of maintenance and provides a helpful, practical structure within which they should be executed.

Andrew Marmorstein
CACI, Inc.

User Interface Design for Computer Systems

Tony Rubin (John Wiley & Sons, New York, 1988, 195 pp., \$39.95)

I continue to be amazed that publishers still print textbooks that contain spelling and grammar errors. This is unconscionable in these days of computerized everything, and especially so when the textbook is about a computer-related subject. *User Interface Design for Computer Systems* is one of those books, and its errors, together with the author's British writing style and inconsistent use of the active and passive voice, contributed to my dislike of the author's treatment of the subject matter.

The author claims that he wrote this book to sensitize readers to issues related to the human-computer interface. If that's all he wanted to do, I suppose he was successful. However, there are more thorough and useful publications that accomplish this and much more.

The book contains eight chapters, but really can be thought of as having four sections. The first five chapters, written by the author himself, contain a pedantic treatment of the nature of users, menus, and input devices. The author's discussion of menus focuses on the Macintosh, and his discussion of input devices focuses on mice, joysticks, trackballs, light pens, touch screens, and keyboards. He brings this all together in a section called "Wimp Interfaces" — windows, icons, menus, and pointing devices.

Unfortunately, this discussion omits other interfaces that have been around for a while and are beginning to be as important as the Mac's. For example, the author does not mention Windows, Presentation Manager, or New Wave. As I mentioned above, his treatment of these subjects is pedantic. This is because he attempts to envelop the discussion in an

The author envelops the discussion in an academic aura, complete with distracting cross references to other chapters.

academic aura, complete with references to other publications and distracting cross references to other chapters. Unfortunately, most of his references are three to five years old. This is not surprising, since much of this material has become commonplace, hence unworthy of more contemporary research.

The second and third sections of the book were written by other authors and

are of significantly higher quality. "Using Color" is an excellent overview, covering the topic from the perspectives of physics, physiology, and engineering. The chapter concludes with useful advice on the use of color for the design of application screens. Both this chapter and the next are replete with up-to-date reference material.

"Evaluating User Interface Designs" essentially reviews the literature of assessment methodologies. While the author contends he is directing the material toward user-interface evaluation, the discussion is actually much less specific than that. This chapter could just as well be describing methodologies for collecting market research information on consumer preferences.

The final chapter — "So You Want To Be A Professional?" — is completely irrelevant considering the title of the book. Except to add a few pages, this material seemed to serve no purpose.

In summary, this is one of those texts you might reference just in case you missed something while looking through other books on interface design. But if you decide to ignore it, you won't have missed much.

Sorel Reisman
California State University, Fullerton

Creating User Interfaces by Demonstration

Brad A. Myers (Academic Press, San Diego, Calif., 1988, 276 pp., \$29.95)

This book is the product of the author's research toward his PhD at the University of Toronto. Issues related to designing user interfaces and building the tools to construct them form the unifying thread throughout the book. The author describes not only the way his experimental system helps designers build interfaces, but also the interface itself, which was built using a user-interface management system called Peridot. Peridot allows a designer to construct the visual aspects of an interface, similar to a paint or draw program; then the designer demonstrates how the system should respond to user inputs and actions.

The book contains 15 chapters and four appendixes. The first two chapters provide the foundations for considering the development of Peridot, including a particularly thorough and well-written chapter on related research. As part of

the coverage of related research, the author describes current work in user-interface management systems, visual programming, and example-based programming, and clarifies the problems related to each of these practices. He then develops a taxonomy classifying prior efforts.

Chapters 3-12 describe in detail the implementation of Peridot's features. Throughout these chapters the author illuminates design decisions that resulted in particular features. A particularly interesting aspect of the system is the implementation of inferencing. The system uses plausible inferencing to establish relationships among graphical objects and mouse dependencies.

The remaining three chapters include considerations for future work, experiences with Peridot, and conclusions. Included in Chapter 14 are the results of an informal experiment using Peridot to

build an interface. Though the data is informal, the results are intriguing. (On a given example, expert programmers using their own hardware averaged three hours to construct the interface, where a novice using Peridot constructed the same interface in 16 minutes.)

I found the book interesting and well written. The author goes beyond the merely descriptive to provide in-depth and consistent explanations of the rationale behind the system. This excellent book is important, fascinating, and intriguing reading for designers interested in user-interface design and human-computer interaction. I highly recommend it for those concerned with practical or research efforts involving the development of quality and effective user interfaces.

Richard Upchurch
Southeastern Massachusetts University

An Introduction to Formal Language Theory

Robert N. Moll, Michael A. Arbib, and A.J. Kfoury (Springer-Verlag, New York, 1988, 203 pp., \$39.90)

Moll, Arbib, and Kfoury have written a concise treatment of formal language theory and computational linguistics, coupled with a history of the field. Their emphasis is on theory, augmented with discussions of selected issues in computational linguistics. Appropriate and significant propositions, theorems, and lemmas are logically presented and methodically proved.

The authors claim the material is self-contained, with little background required. In truth, most readers would need a significant knowledge of set theory, and some recollection of matrix algebra, automata, and related topics, to readily comprehend and follow the discussion. Including a primer on notation and concepts would help either as a review or for the uninitiated.

The writers' choice of symbols departs from the norm, making it tedious at times

specific to the final two chapters. References at the close of each chapter would inspire additional reading.

However, the book has its strengths. Some of the individual topic explanations are the best I have seen. Such topics include the decomposition of trees and detailed parsing mechanisms, which have often been treated in far less detail in other works. Push-down automata and Turing machines also are especially well covered.

The authors develop most of the proofs meticulously and completely. However, as noted before, to fully appreciate the proofs, the reader would need either a firm grasp of set theory and general mathematics or suitable references at hand. Some proofs, though, are left to the reader to develop.

After each section, the authors provide numerous exercises of varying complex-

The section on fixed-point theory presents the first algebraic approach to context-free languages I have seen.

to compare selected material with other works. The most notable example is the use of λ to represent the null or empty string, as opposed to the much more prevalent ϵ . Often, symbols are used without a complete or adequate explanation, only to be covered much later, if at all. The symbol index is incomplete and of little use.

As a text, this book is probably most suitable for advanced graduate students, such as those in a PhD program. If used in an undergraduate or beginning graduate course, as the book's preface suggests, much supplemental explanation and clarification would be needed. In fact, the authors refer to their other works as companion or complementary volumes. These are *A Basis for Theoretical Computer Science*, *A Programming Approach to Computability*, and *Algebraic Approaches to Program Semantics*, all published by Springer-Verlag.

It should be noted that the authors succinctly cover in 195 pages material to which others have devoted 500 or more pages. Given such a condensed though exacting treatment, substantial elaboration in an educational setting should be expected.

Another shortcoming is the lack of a bibliography, except for 17 references

ity. These meaningful exercises are worth the time and effort, and they reinforce the material just communicated.

While much of the material in the first seven chapters has been treated elsewhere, the final two chapters provide a discussion and review of natural languages and some new approaches to linguistic theory. This material is current and was prepared principally by contributing author James Pustejovsky.

Major topics overall include grammars and languages, closure, push-down automata, equivalence theorem, parsing, Turing machines, halting problem and undecidability, fixed-point principles, generative power of natural languages, augmented transition networks, and phase structure grammars. The section on fixed-point theory presents the first algebraic approach to context-free languages I have seen and is particularly meritorious.

I certainly recommend this book as an excellent reference or as a text in an advanced course focusing on very formal or theoretical developments where the participants have a sufficient and current background in the underlying mathematics.

Richard G. Estock
EDP Consultants

Neural Networks and Natural Intelligence

Stephen Grossberg, ed. (MIT Press, Cambridge, Mass., 1988, 637 pp., \$35)

This book consists of reprints of 12 research articles written by Steven Grossberg alone or jointly with one of 10 other researchers from Boston University's Center for Adaptive Systems. A computing book club has offered it for sale, giving the impression, incorrectly, that the material is appropriate for computing professionals keeping abreast of novel developments in AI. In fact, Grossberg states in his introduction a "3/4 Rule" describing the center: "Each staff member at the CAS typically has training in at least three of the four fields: computer science, mathematics, psychology, and neurobiology." 2/4-people are guaranteed a significant struggle with the material; heaven help the 1/4-ers.

The articles use mathematical models (often expressed as systems of differential equations) and experiments by computer simulation to explore a wide range of topics within psychology and neurophysiology, including vision (form, color and brightness perception, stereopsis), emergent segmentation, attention, competitive learning, conditioning, planned motor control, and decision making under risk.

The unifying principle in these studies is Grossberg's Adaptive Resonance Theory (ART), a neural network model that captures and significantly enhances the statistical technique known as cluster analysis. Cluster analysis generalizes the notion of multimodality, partitioning a collection of data into subsets consisting of similar objects ("similar" is domain specific). Also known as numerical taxonomy, clustering is a form of unsupervised "learning" that makes sense of and imposes structure on a mass of data. This forms a mathematical abstraction of what living beings do when they group sensory data into objects (as an infant learns to track moving objects with its eyes), recognize a person today as the same as yesterday, and form concepts.

Standard techniques for cluster analysis are popular candidates for parallelization and implementation as neural network models, but unmodified they provide poor mechanisms for explaining or mimicking natural systems. Cluster analysis algorithms cannot function unassisted without prior knowledge of some aspects of the form and structure of their data. They cannot cope with raw, surprising, changing data spaces; the data clusters might merge when they should not, or two "natural clusters"

might separate. Many artificial neural network models similarly overload and fail to learn and recall.

ART incorporates mechanisms such as attention, multiple interacting memories, and a modifiable vigilance system to overcome what Grossberg calls the stability-plasticity dilemma. For example, the centroid of a cluster or the long-term memory representative of a classification interacts with a sensory pattern newly presented to the system, and misfit parameters are evaluated that will accept or reject the proposed classification, perhaps updating the representative centroid (a conceptual code), perhaps initiating a

new classification.

Grossberg presents ART in detail in the chapter, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," coauthored with Gail A. Carpenter. Carpenter and Grossberg also presented ART in a more introductory article in *Computer's* March 1988 special issue on artificial neural systems. This article can provide an overview and "ramping up" for the eventual study of the difficult book under review.

Peter G. Anderson
Rochester Institute of Technology

NEW LITERATURE

IEEE salary survey. The IEEE is accepting orders for the *1989 IEEE US Member Salary and Fringe Benefit Survey*. The guide explores more than 50 variables affecting salaries and benefits, highlighting retirement planning and pension practices, early retirement issues, and salary progression.

For orders placed before May 15, the price is \$47.95 for IEEE members and \$67.95 for nonmembers. The price after May 15 is \$59.95 for members and \$79.95 for nonmembers.

Contact the IEEE Service Center, 445 Hoes Lane, PO Box 1331, Piscataway, N.J. 08855-1331, phone (201) 981-1393. Specify IEEE catalog number UH0183-4.

Object-oriented-computing tutorials. The IEEE Computer Society Press has recently issued two new tutorials dealing with the increasingly important field of object-oriented computing. *Object-Oriented Computing, Volume 1: Concepts* (ISBN 0-8186-0821-8, 215 pp., softbound, \$25 members, \$40 nonmembers) introduces the basic concepts in the field on a level designed for a practicing computer engineer. Author Gerald Peterson emphasizes the Smalltalk and Ada languages, but also discusses object-oriented aspects of such languages as Modula-2 and Lisp.

Object-Oriented Computing, Volume 2: Implementations (ISBN 0-8186-0822-6, 320 pp., softbound, \$30 members, \$50 nonmembers) offers a thorough summary of techniques used so far. The book examines object-oriented design, databases, and architectures.

Contact the IEEE Computer Society Press, 10662 Los Vaqueros Circle, Los Alamitos, Calif., 90720-2578, phone (714) 821-8380.

Robotics proceedings. The proceedings of the Second International Symposium on Robotics have been published by the American Society of Mechanical Engineers as *Robotics and Manufacturing: Recent Trends in Research, Education, and Applications* (book number 800067, \$132 to ASME members, \$165 to nonmembers). The volume contains about 150 papers in 18 categories, including control architecture, intelligent control, mobile and flexible robots, and robots in manufacturing.

Contact the ASME Order Department, 22 Law Drive, Box 2300, Fairfield, N.J. 07007-2300, phone (800) 843-2763.

Protecting ideas. The newly published third edition of *Protecting Engineering Ideas and Inventions* (softcover: ISBN 0-944606-04-0, \$46; hardcover: ISBN 0-944606-05-9, \$54) by Ramon D. Foltz and Thomas A. Penn has updated information and more than 75 additional pages of new material. The volume covers such topics as patents, copyrights, trademarks, trade secrets, secrecy agreements, licensing, and warranties. Contact the Penn Institute, PO Box 41016, Cleveland, Oh 44141, phone (800) 426-7495.

R&D advice. Robert Szakonyi includes the viewpoints of 10 veteran R&D managers in *How to Successfully Keep R&D Projects on Track* (ISBN 0-912338-25-3, \$29). Among the contributors are veterans of Marietta Aerospace, Fairchild Industries, Battelle Institute, Forecasting International, the Naval Ship R&D Center, and the National Institutes of Health. Contact Lomond Publications, PO Box 88A, Mt. Airy, Md. 21771, phone (800) 443-6299.

ASEE Faculty Update Courses

Design of Telecommunication Networks

Polytechnic University
July 10-21, 1989

One of several courses sponsored by the American Society for Engineering Education to help professors to update their senior/graduate courses. Others on Work Stations, VLSI Design, Robotics, Lasers, UNIX/C, Digital Signal Processing, at Geo. Tech, MIT, RPI, Rice, AT&T, IBM. Small interactive classes of faculty-participants, including hands-on computer work, notes and software. Apply by March 17, 1989. For more information, contact:

Ms. Jean King
ASEE
11 Dupont Circle, NW, Suite 200
Washington, DC 20036
Phone: (202) 293-7080

Late Magazines?
No Magazines?
Membership
Status Problems?
No Answers
To Your
Complaints?

Let your
Computer
Society
Ombudsman
cut
through
the red
tape
for you.



THE COMPUTER SOCIETY
Ombudsman
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
COMPMAIL + : CS.HELP

ADVERTISER INDEX

American Society for Engineering Education	118
Boston University College of Engineering	114
CACI Products Company	1, Cover IV
CASE '89	40
Computer Architecture and Digital Signal Processing Symposium	71
Computer Society Membership	106, 115
Computer Society Ombudsman	119
Computer Society Press	97
Data Engineering Conference	107
ICCAD '89	Cover II
Guidelines Software, Inc.	19
Institute for Defense Analyses	73
Integrated Network Management Symposium	89
Martin Marietta Laboratories	78
New Jersey Institute of Technology	114
Northrop Defense Systems Division	61
Oakland Group, Inc.	8
Parallel Processing Symposium	106
Quotron Systems, Inc.	20
Security and Privacy Symposium	4-5
Software for Strategic Systems Conference	48
STARS Conference	42
Supercomputing '89	79
TASC	48, 103
Tools for AI Workshop	80
Visible Systems Corp.	30
Visualization in Biomedical Computing Conference	62
Classified Advertising	108-114

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

Northern California and Pacific Northwest: Roy McDonald Assoc. Inc., 5915 Hollis St., Emeryville, CA 94608; (415) 653-2122.
Jim Olsen, P.O. Box 696, Hillsboro, OR 97123; (503) 640-2011.
Southern California and Mountain States: Richard C. Faust Co., 24050 Madison St., Suite 100, Torrance, CA 90505; (213) 373-9604.
Southwest: The House Co., 3000 Wesleyan, Suite 345, Houston, TX 77027; (713) 622-2868.
Midwest: Robert Acker & Assoc., 480 Central Ave., Northfield, IL 60093; (312) 441-6050.
East Coast: Atlantic Representative Group, 349 Maple Pl., Keyport, NJ 07735; (201) 739-1444.
New England: Arpin Associates, 40 Sterling St., Somerville, MA 02144; (617) 625-1777.
Europe: Heinz J. Görgens, Parkstrasse 8a, D-4054 Nettetal 1 - Hinsbeck (F.R.G.); phone: (0 21 53) 8 99 88; telex 841(17)2153310=HJG tlx d.

Advertising Director: Dawn Peck. For production information, conference, and classified advertising, contact Heidi Rex or Marian Tibayan.

COMPUTER, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; phone (714) 821-8380; fax (714) 821-4010.

PRODUCT INDEX

	RS#	Page#
CIRCUITS		
IC announcements	120-133	95
COMPONENTS		
Microsystem announcements	135-148	96
CONFERENCES & COURSES		
CA-DSP '89	—	71
CASE '89	—	40
Data Engineering	—	107
ICCAD '89	—	C.II
Integrated Network Management	—	89
Parallel Processing	—	106
Security and Privacy	—	4-5
STARS	—	42
Strategic Systems	—	48
Supercomputing '89	—	79
Telecommunication Networks	—	118
Tools for AI	—	80
Visualization in Biomedical Computing	—	6
DATA ACQUISITION		
Portable recorder	23	87
Signal processing software	22	85
Software package	20	81
DATA COMMUNICATIONS		
Development kit	34	91
PUBLISHERS		
CS membership	—	106, 115
CS ombudsman	—	119
CS Press	—	97
RECRUITMENT		
Classified advertising	—	108-114
Technical professional	20, 48, 61, 73, 78, 103, 114	
SOFTWARE		
Ada repository	38	92
C compiler	2, 46	19, 94
CADD	32	90
CASE tool	42, 47, 50	93, 94
Project estimator	33	90
SYSTEMS		
Automation operating system	45	93
CAD system	49	94
Computer	30-31, 35-37, 40	90-92
Development system	48	94
Management system	1	8
Productivity tool	3	30
Simulation package	41	1, 93, C.IV
Vision system	43, 44	93
Viewing/annotation system	39	92
TEST & MEASURING EQUIPMENT		
Laboratory instrumentation	21	84

COMPUTER**March 1989 issue** (Card void after August 1989)

Please print or use peel-off address label.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest(Circle what you liked;
add comments on the back)

- A** Guest Editor's Intro
B Motion Planning
C Task-Level Planning
D Geometric Computation
E Exception Handling
F Research Overviews
G Query Processing

- H** Standards
J Computer Society News
K Update
L Product Reviews
M New Products
N IC Announcements
P Microsystems
Q Conferences
R Calendar
S Call for Papers
T Book Reviews
U New Literature

Product inquiries(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

1**COMPUTER****March 1989 issue** (Card void after August 1989)

Please print or use peel-off address label.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest(Circle what you liked;
add comments on the back)

- A** Guest Editor's Intro
B Motion Planning
C Task-Level Planning
D Geometric Computation
E Exception Handling
F Research Overviews
G Query Processing

- H** Standards
J Computer Society News
K Update
L Product Reviews
M New Products
N IC Announcements
P Microsystems
Q Conferences
R Calendar
S Call for Papers
T Book Reviews
U New Literature

Product inquiries(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

2**COMPUTER****March 1989 issue** (Card void after August 1989)

Please print or use peel-off address label.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest(Circle what you liked;
add comments on the back)

- A** Guest Editor's Intro
B Motion Planning
C Task-Level Planning
D Geometric Computation
E Exception Handling
F Research Overviews
G Query Processing

- H** Standards
J Computer Society News
K Update
L Product Reviews
M New Products
N IC Announcements
P Microsystems
Q Conferences
R Calendar
S Call for Papers
T Book Reviews
U New Literature

Product inquiries(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

3

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader-service inquiries, see other side.



PO Box is for reader-service cards only.

PLACE
POSTAGE
HERE

COMPUTER

Reader Service Inquiries

PO Box 16508

North Hollywood, CA 91615-6508

USA



Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader-service inquiries, see other side.



PO Box is for reader-service cards only.

PLACE
POSTAGE
HERE

COMPUTER

Reader Service Inquiries

PO Box 16508

North Hollywood, CA 91615-6508

USA



Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader-service inquiries, see other side.



PO Box is for reader-service cards only.

PLACE
POSTAGE
HERE

COMPUTER

Reader Service Inquiries

PO Box 16508

North Hollywood, CA 91615-6508

USA





IEEE COMPUTER SOCIETY

A member society of the Institute of Electrical and Electronics Engineers, Inc.

Executive Committee

President: Kenneth R. Anderson*
Siemens Research & Technology
755 College Road East
Princeton, NJ 08540
(609) 734-6550

President-Elect: Helen M. Wood*
Past President: Edward A. Parrish, Jr.*

Vice Presidents

Conferences and Tutorials: Joseph E. Urban (1st VP)*
Technical Activities: Laurel V. Kaleda (2nd VP)*

Area Activities: Ned Kornfield†
Education: Gerald L. Engel†

Membership and Information: Barry W. Johnson†

Press Activities: Duncan H. Lawrie*

Publications: Sallie V. Sheppard*
Standards: Paul L. Borrill*

Secretary: Michael Evangelist*

Treasurer: Charles B. Silio†

Division V Director: Harriett Rigas†

Division VIII Director: Roy L. Russo†

Executive Director: T. Michael Elliott†

*voting member of the Board of Governors

†nonvoting member of the Board of Governors

Board of Governors

Term Expiring 1989:

Bill D. Carroll, Lansing (Chip) Hatfield,
Duncan H. Lawrie, David Pessel,
Susan L. Rosenbaum, Sallie V. Sheppard, Bruce Shriver,
Harold S. Stone, Akihiko Yamada, Marshall C. Yovits

Term Expiring 1990:

Vishwani Agrawal, Mario R. Barbacci,
Ming T. (Mike) Liu, Yale N. Patt, Donald E. Thomas,
Benjamin W. Wah, Ronald Waxman

Term Expiring 1991:

P. Bruce Berra, Paul L. Borrill, Michael Evangelist,
Ted Lewis, Raymond E. Miller,
Earl E. Swartzlander, Jr., Thomas W. Williams

Next Board Meeting

May 19, 1989: 8:30 a.m.
Vista International Hotel, Pittsburgh

Senior Staff

Executive Director: T. Michael Elliott
Editor and Publisher: H. True Seaborn
Publisher, Computer Society Press: Eugene M. Falken
Director, Conferences and Tutorials: Anne Marie Kelly
Director, Finance and Administration: Tod S. Heisler
Assistant to the Executive Director: Violet S. Doan

Computer Society Offices

Headquarters Office

1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Telex: 7180250437 IEEE COMPSO
Fax: (202) 728-9614

Publications Office

10662 Los Vaqueros Cir.
Los Alamitos, CA 90720-2578
Membership and General Information: (714) 821-8380
Publication Orders: (800) 272-6657
Fax: (714) 821-4010

European Office

13, Ave. de L'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Telex: 25387 AVVALB
Fax: 32 (2) 770-85-05

Asian Office

Ooshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan
Phone: 81 (3) 408-3118
Fax: 81 (3) 408-3553

Use the Reader Service Card to obtain information on:

- Membership application—student #203, others #202
- Periodicals subscription form for individuals #200
- Periodicals subscription form for organizations #199
- Publications catalog #201
- Standards working groups list #195
- Comppmail+ international electronic mail/database brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures—student/regular #193
- Student scholarship information #192
- Awards description/nomination forms #198
- Volunteer leaders/staff directory #196
- IEEE senior member application #204

Purpose

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 97,000 members worldwide, and provides a wide range of services to members and nonmembers.

Membership

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others seriously interested in the computer field.

Publications and Activities

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and new products.

Periodicals. The society publishes six magazines and four research transactions. Refer to membership application or request information as noted above.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. Over 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. More than 30 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

European Office

Payments for Computer Society membership and publication orders are accepted by checks in Belgian, British, German, Japanese, Swiss, or US currency. Checks in US funds must be drawn on a US bank. Payment may also be made by American Express, Diners Club, Eurocard, Master Card, or Visa credit cards.

Asian Office

Payments for Computer Society membership and publication orders are accepted by checks in Japanese or US currency. Checks in US funds must be drawn on a US bank. Payment may also be made by electronic fund transfer to the Bank of Tokyo, Akasaka Branch, Toza acct. 0767956; the credit receiver is the IEEE Computer Society Headquarters Office. Payment may also be made by American Express, Diners Club, Eurocard, Master Card, or Visa credit cards.

Ombudsman

Members experiencing problems — magazine delivery, membership status, or unresolved complaints — may write to the ombudsman at the Publications Office.

Breakthrough in presentation of simulation results

SIMSCRIPT II.5 with SIMGRAPHICS

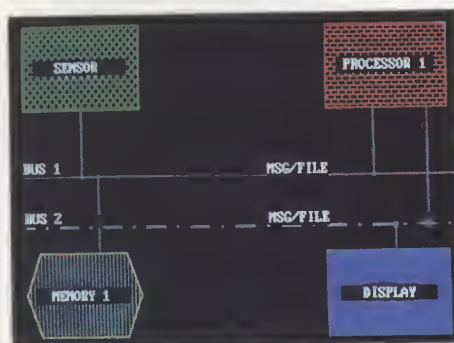
Now you see an animated picture of the system



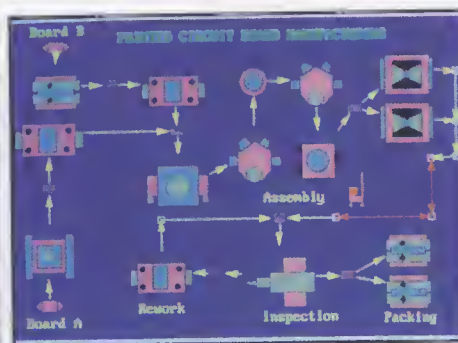
SIMGRAPHICS™ scenario preparation



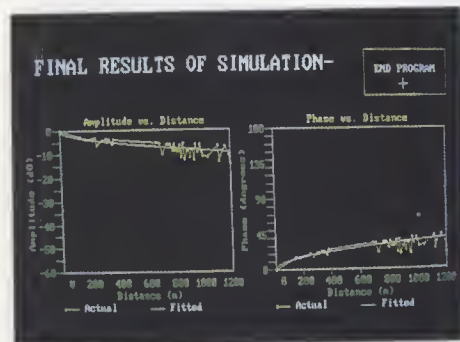
Use your icons--or ours



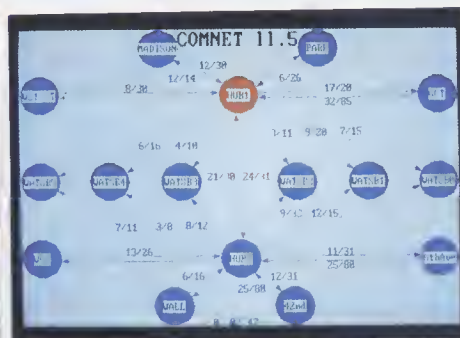
Computer communications-NETWORK II.5®



Factory in operation-SIMFACTORY®



Presentation graphics



Telecommunications-COMNET II.5®



Map with moving objects



SIMGRAPHICS menu builder

Free trial and training

See for yourself how simulation results are now easier to understand.

The free trial contains everything you need to try SIMGRAPHICS™ on your computer.

We send you SIMSCRIPT II.5, animated models, and complete documentation. You can build your own model or modify one of ours.

Try the SIMSCRIPT II.5® language, the timeliness of our support, the accuracy of our documentation, and the facilities for error checking--everything you need for a successful project.

For 26 years CACI has offered trial use of its simulation software--no cost, no obligation.

Act now for free training

For a limited time we will also include free training.

For immediate information

Call Hal Duncan at (619) 457-9681. In the UK, call Richard Eve on (01) 528-7980.

Rush information on SIMSCRIPT II.5 with SIMGRAPHICS

Limited offer--return the coupon today and we will also include one free course enrollment worth \$950.

☐ Send information on your Special University Offer.

Name _____
Organization _____
Address _____
City _____ State _____ Zip _____
Telephone _____
Computer _____ Operating System _____

Return to:
CACI Products Company
3344 North Torrey Pines Court
La Jolla, California 92037
Call Hal Duncan at (619) 457-9681.
FAX (619) 457-1184.

In the UK:
CACI Products Division
Regent House, 89 Kingsway
London, WC2B 6RH, United Kingdom
Call Richard Eve on (01) 528-7980.

SIMSCRIPT II.5, COMNET II.5, NETWORK II.5, and SIMGRAPHICS are registered trademarks and service marks of CACI, INC.
©1989 CACI, INC.